Tomographic determination of interval velocities from picked seismic data—theory and synthetic results

Chuck Sword

ABSTRACT

The method of Controlled Directional Reception (CDR) can be used to determine travel times and ray parameters of waves transmitted from a given shot and received at a given geophone (Riabinkin et al., 1962; Sword, 1984). These ray and travel-time parameters (collectively known as CDR parameters) are obtained from conventional seismic data through an automated picking procedure, and can be inverted tomographically to give the velocity structure of the medium. No assumptions are made as to the shape or continuity of the reflecting horizons. The inversion process has been successfully tested on synthetic data from a laterally heterogeneous model with curved, dipping reflectors.

INTRODUCTION

In a recent paper Bishop et al. (1985) discuss the determination of interval velocities, and they give a concise summary of various methods that have been used in the past. They then describe a tomographic approach, where picked travel-time data are inverted in order to determine interval velocities and depths of reflecting horizons.

My present paper also describes a tomographic approach. One major difference between my approach and that of Bishop et al. is that rather than use travel time picks from interpreted horizons, my approach utilizes travel times and ray parameters that can be picked automatically from common-shot and common-geophone gathers (the automated picking of such parameters is the basis of the method of Controlled Directional Reception (CDR), which has been developed over the last several decades in the Soviet Union). This approach is similar to those previously proposed by Harlan and Burridge (1983) and by Gray and Golden (1983). Only interval velocities are inverted for, not depths of reflectors. Because of the differences between this method and that of Bishop et al., a somewhat different approach to the inversion problem is necessary.

Inverse problems are usually formulated in terms of a search for the minimum of an objective

function, where the objective function describes the mismatch between the real data and the data predicted by a model. Typically, tomographic velocity inversions seek to minimize the discrepancy between observed and predicted travel times (see, for instance, Worthington (1984)). I have chosen a somewhat unusual objective function. Rather than trying to minimize the error in predicted versus measured travel time, the method that I present here seeks to minimize the horizontal distance between the endpoints of each pair of downgoing and upcoming rays.

Some tomographic methods operate under the assumption that the rays travel in straight lines; I, like Bishop et al. (1985), have chosen to allow curvature of rays. Often it is possible to use Fermat's principle to justify using the same raypaths over several iterations, even though the velocity may have changed (Toldi, 1985); my objective function is such that Fermat's principle cannot be invoked.

Once an objective function has been chosen, a procedure must be defined that will minimize this function. I use a technique known as the Gauss-Newton method (Gill et al., 1981; Levin, 1985), which works as follows. An initial velocity model is chosen, and the rays that correspond to the various travel-time and ray-parameter picks are traced through this model in order to find the value of the objective function. The results are used to form a linearized least-squares inversion problem. This least-squares problem can be solved by the conjugate-gradient method (e.g., with the subroutine LSQR (Paige and Saunders, 1982)); the solution is used to update the velocity model. Rays are then traced in the new velocity model, and the sequence of iterations continues.

This inversion scheme has been tested on synthetic data. The picked parameters were generated directly by the modeling programs, rather than being picked from synthetic field profiles. The tests were successful, although with one of the synthetic data sets there was some difficulty in determining the interval velocities in the lower part of the model.

There are several advantages to the tomographic method proposed in the present paper. No assumptions are made about the shape or continuity of the reflectors, and data obtained from point diffractors will work as well as data from flat, dipping, curved, or faulted horizons. The picking is automated, so there is no need to interpret horizons and digitize travel times manually. The time and ray-parameter picks should occupy much less space than the original data set, producing savings in both processing time and data storage.

There are also some disadvantages. In contrast to the method proposed by Bishop et al., the present method is not based on the implicit assumption that reflectors are continuous; as a result, a certain amount of robustness is lost (the "focusing" of horizons is not used as a criterion in the inversion). Since the data are picked automatically, the present method may be adversely affected by multiples and other coherent noise that a human interpreter could easily exclude. In contrast to Fowler's proposed method (1985), which is based to some extent on Toldi's work (1985), the present method expends a great deal of computer time picking reflections and tracing rays. There

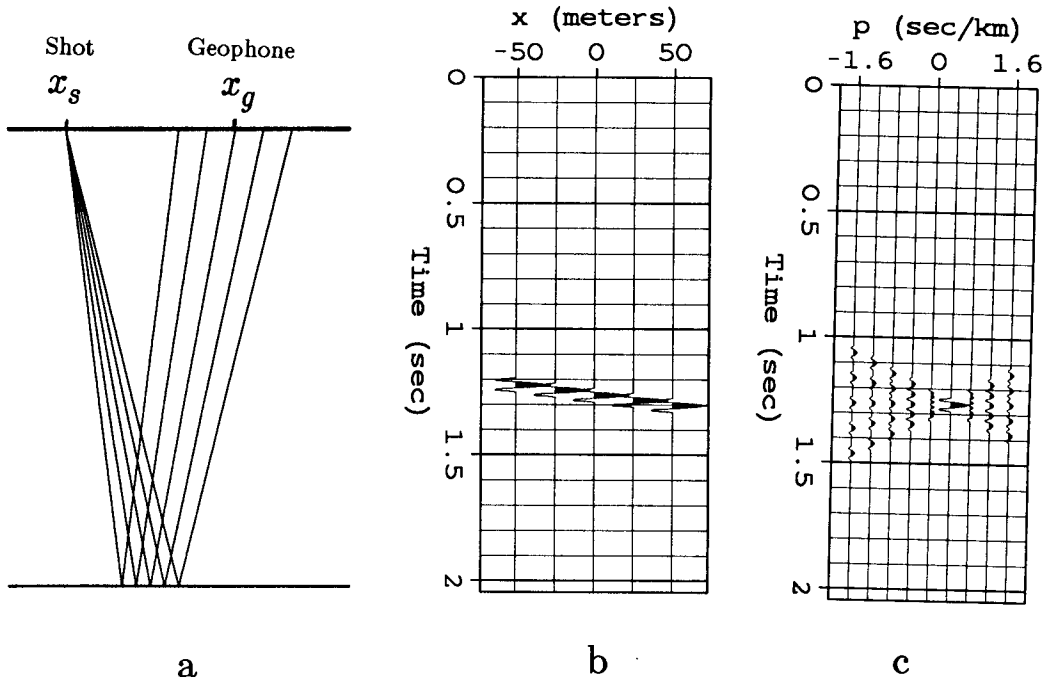


Figure 1: Determining ray parameters. Figure 1a shows how a short-base common-shot gather might be collected in the field. Figure 1b shows a possible outcome of this seismic experiment (notice that the horizontal coordinate in this Figure has its origin at the location x_g). Figure 1c shows the result of slant stacking this data; from the slant-stack section it is possible to determine the ray parameter p_g (defined as $\Delta t/\Delta x_g$) and travel time t.

is also the general problem inherent in any method where rays are traced—rays are fickle, and the smallest changes in velocity can cause them to shoot off in unpredictable directions. Solutions to the wave equation may be much more stable in this regard.

SUMMARY OF CDR (CONTROLLED DIRECTIONAL RECEPTION)

The method of Controlled Directional Reception (CDR) was invented in the United States, but reached its highest development in the Soviet Union (Riabinkin et al., 1962; Hermont, 1979; Sword, 1984). The CDR approach might best be summarized by the phrase, "slant stacks over short bases". As illustrated in Figure 1, a portion of a common-shot or common-geophone gather is transformed into a slant-stack section. The peaks of the slant stack are picked, yielding the travel times and ray parameters of the various events in the section. In practice, it is preferable to use a semblance-weighted slant stack (Stoffa et al., 1981; Kong et al., 1985), as the peaks are sharper and more easily picked, and aliased noise (such as ground roll) is better attenuated.

Figure 2 shows a typical common-shot gather (a field profile). A portion of this gather was slant stacked and semblance weighted, with the result shown in Figure 3a. The peaks picked

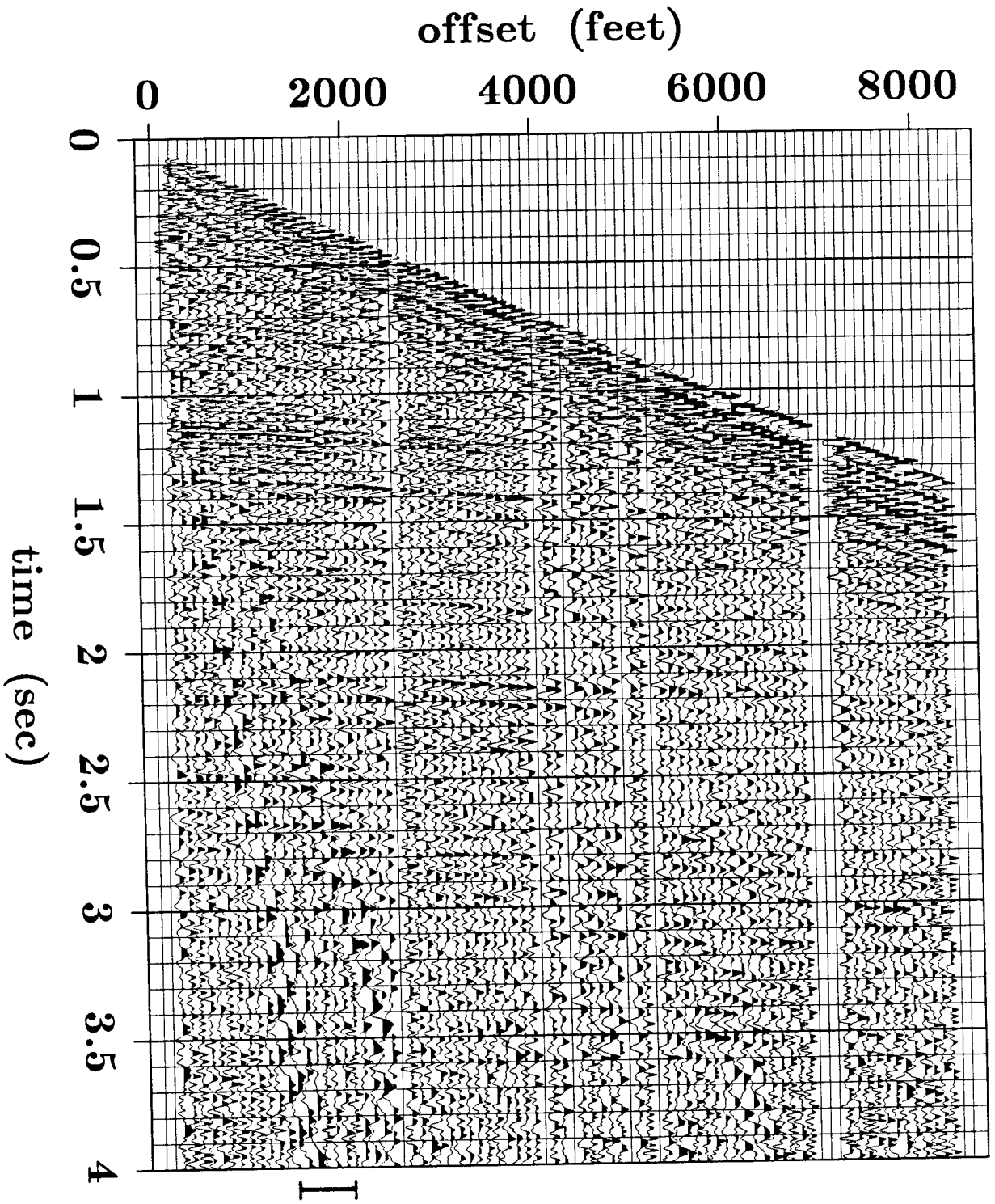


Figure 2: A common-shotpoint gather, courtesy of an anonymous sponsor. Offsets 1400–2000 (denoted by the heavy line at the bottom of the figure) will be used to produce Figure 3. The entire gather will be used to illustrate the automated picking of CDR parameters.

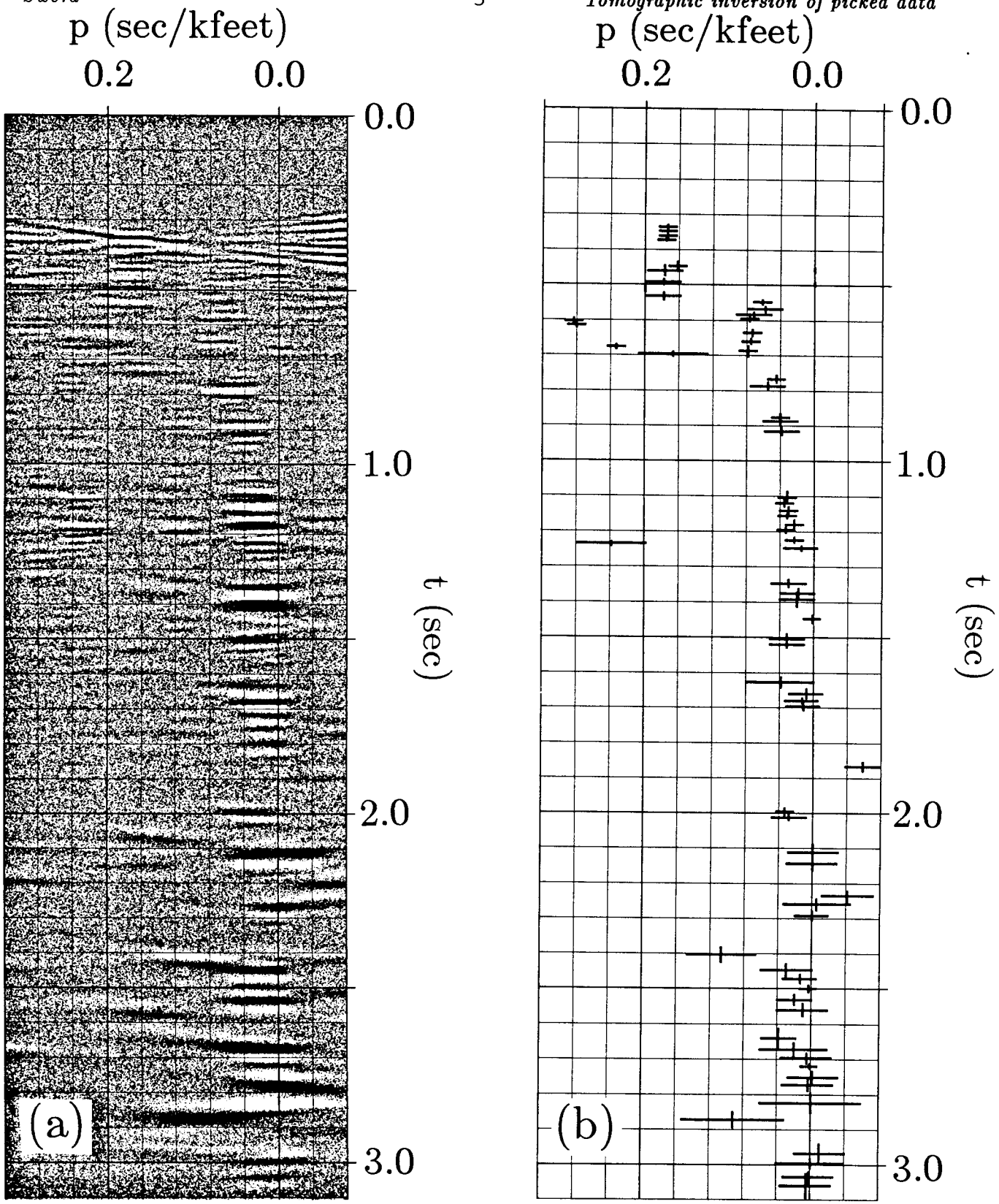


Figure 3: Semblance slant stack and automatic picking. Offsets 1400–2000 in Figure 2 were slant stacked and semblance weighted to produce the data illustrated in Figure 3a. Figure 3b shows the peaks that were picked by an automatic picking algorithm.

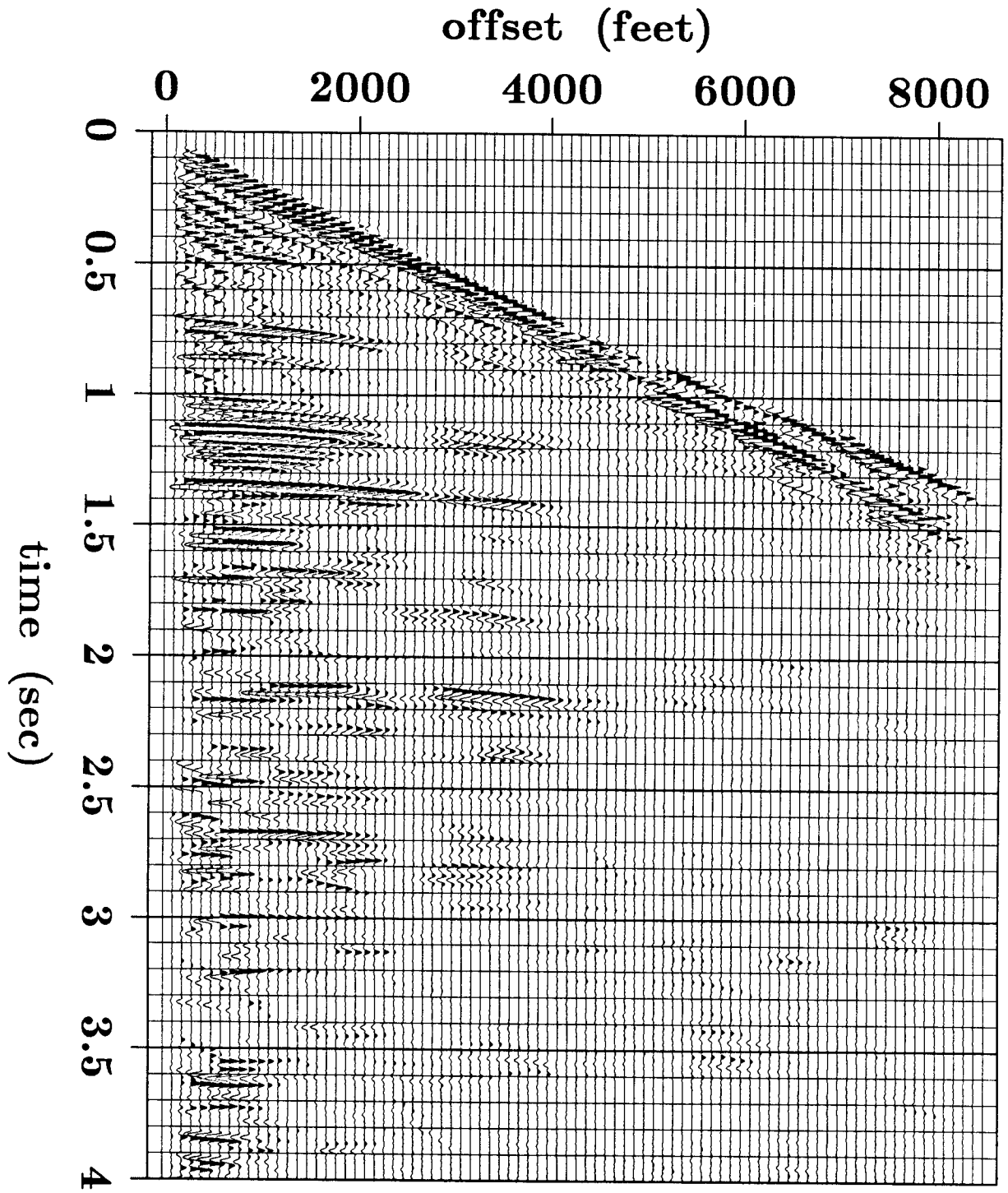


Figure 4: A synthetic time section. The data in Figure 2 were slant stacked and automatically picked, as illustrated in Figure 3. The picks were used to reconstruct the original data, with the results shown here. The picking algorithm managed to find the main events, but it has picked up some noise as well.

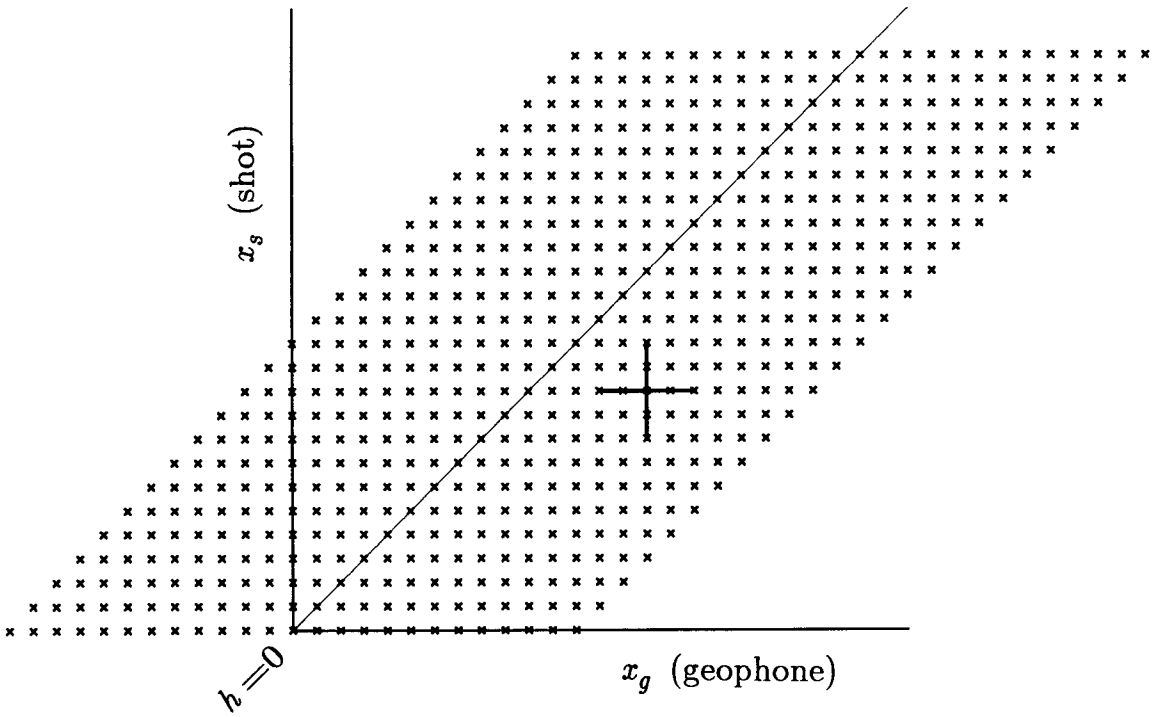


Figure 5: A stacking chart showing the "summation bases" that might be used to determine p_s and p_g at a particular x_s and x_g. The heavy horizontal line denotes a short-base common-shot gather, whose traces can be slant stacked and picked in order to determine p_g. The heavy vertical line denotes a short-base common-geophone gather, whose traces can be slant stacked and picked in order to determine p_s.

by a simple-minded automatic picking algorithm are shown in Figure 3b (the picking algorithm is loosely based on ones described by Rapoport (1977)). The entire gather from Figure 2 was processed in this manner: seven adjacent traces were slant stacked, their peaks were picked, the base of summation was shifted by one trace, and the process was repeated. The use of a rolling slant stack (Ottolini, 1983) made the stacking program fairly efficient. Once the peaks were picked, their corresponding ray parameters, travel times, and average shot-geophone distances were stored. These stored parameters were then used to create a synthetic time section: each pick was interpreted as representing a short "dip bar" on the original time section, and a corresponding dip bar was drawn on the synthetic time section. The synthetic time section is shown in Figure 4; it agrees fairly well with the original time section, and demonstrates that the automatic picking algorithm was successful in finding the major events. Unfortunately, the algorithm was also successful at extracting non-existent events from noise. Clearly, more work remains to be done.

Once a portion of a particular common-shot gather has been picked (the portion representing data collected near a particular geophone), the corresponding portion of the associated common-geophone section can be picked (see Figure 5). The picked parameters can be combined to give

travel times, downgoing ray parameters, and upgoing ray parameters of all waves that were produced by that particular shot and that arrived at that particular geophone. This correlation process can be carried out for all shot-geophone pairs, yielding a fairly substantial number of picked parameters. These picked parameters are the input data for the inversion processes that will be described below.

The input data for the inversion process consist of many sets of t, p_s and p_g picks, where t represents the observed travel time, and p_s and p_g represent the downgoing and upcoming ray parameters, of a particular wave. Each set of p_s, p_g, and t has associated with it a value of x_s, the position of the shot where p_s was measured, and x_g, the position of the geophone where p_g was measured. Of these five parameters, x_s and x_g (the shot and geophone position) are of course the most reliable, and t is almost as reliable. The remaining parameters, p_s and p_g, may not be especially accurate. If the velocity of the medium were assumed to be constant, then these five parameters, if accurate, would be sufficient to determine that velocity (Sword, 1984).

OBJECTIVE FUNCTIONS

A possible objective function

Given the five picked parameters and a proposed velocity model, there are several different objective functions that could be used to test how well the model fits the data. The most straightforward one is as follows: trace rays, with ray parameters p_s and p_g, from points x_s and x_g respectively. Keep going until the rays meet. Calculate the total predicted travel time. The objective function consists of the squared difference between the predicted and observed travel times.

This straightforward method is not the one that I chose to use. Figure 6 illustrates a couple of the reasons why. Suppose that the velocity model happens to be correct. Since the model is correct, the objective function should be at a minimum. Suppose also that x_s and x_g are near each other, and that there are slight errors in p_s and p_g. The solid line in Figure 6 shows the actual raypath in the medium, and the dotted line shows the raypath that is predicted on the basis of the erroneous measurements of p_s and p_g. The measured travel time is based on the travel time along the solid line, while the predicted travel time is based on the travel time along the dotted line. So as a result of a small error in measuring the ray parameters, there is a large error in the predicted travel time, which in turn incorrectly suggests that there is a large error in the velocity model. The first problem, then, is that small ray-parameter errors (and these are unavoidable) can lead to large errors in velocity measurement. The second problem has to do with the way the velocity model is updated. Typically in tomographic problems the velocity will be updated along raypaths where a velocity error has been detected. This means that in the example depicted

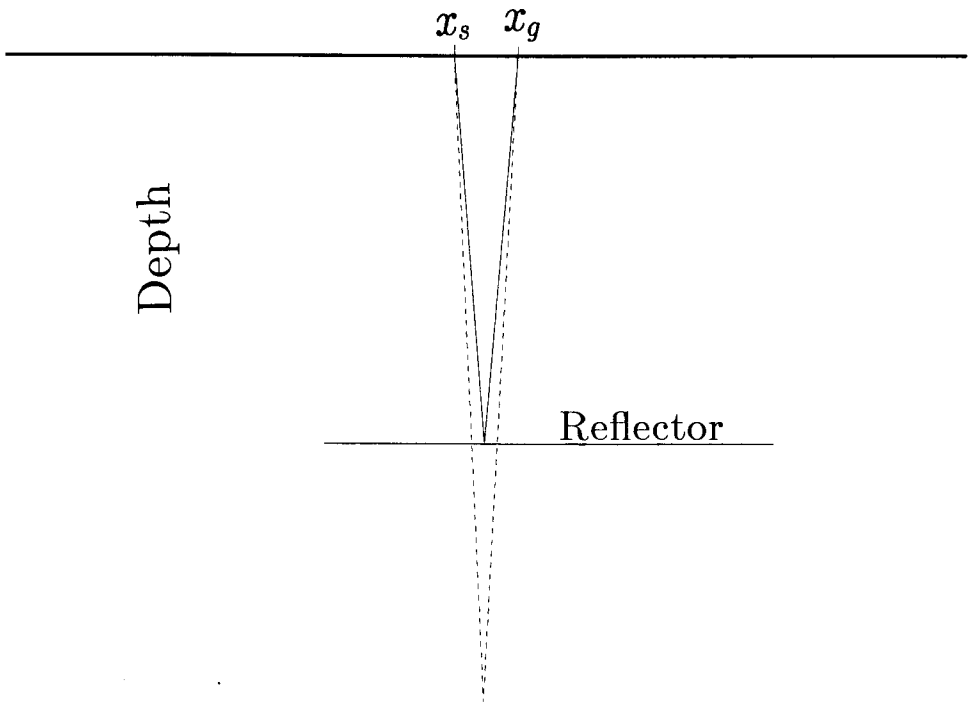


Figure 6: The result of small errors in p_s and p_g. When x_s and x_g are near each other, small errors in the measurement of the ray parameters can lead to large errors in the predicted travel time. Here the solid line shows the actual raypath in the medium, while the dotted line shows the raypath computed on the basis of slightly erroneous ray-parameter measurements.

in Figure 6, the velocity will be updated not only from the surface to the reflector depth, as it should be, but also from the reflector all the way down to the point where the two erroneous rays intersect. It appears, then, that travel-time error is not a good measurement to use in the objective function.

There is yet another reason for not trying to minimize errors in travel time. In tomographic inversion problems, one does not find an objective function that minimizes, for instance, the error in x_s, the shot position—the shot position is assumed to be known exactly. Instead, one tries to minimize the error in another parameter (t, perhaps) that is more likely to be erroneous. In a similar fashion, when the five-parameter CDR data sets described in this paper are being used, there is no sense in trying to minimize the error in the parameters x_s, x_g, or t, which are assumed to be known relatively precisely. The objective function should be based on errors in the ray parameters p_s and p_g, not on errors in the measured versus the predicted travel times.

There are difficulties, however, in basing an objective function directly on errors in the ray parameters. I won't go into details, but it should not be too difficult to see that to measure these errors, an excessive amount of ray tracing would have to be performed. Since t would be

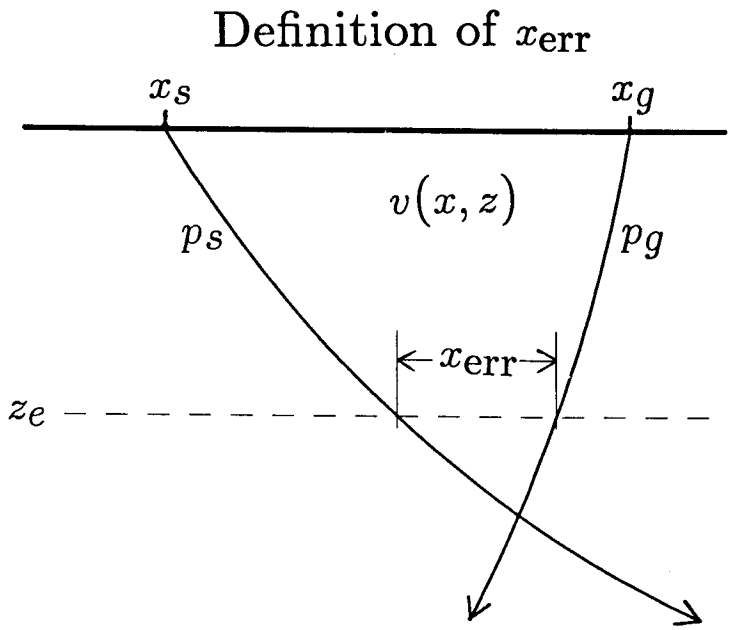


Figure 7: A better objective function. Two rays are traced down to a depth z_e, the depth where their combined computed travel time equals the measured (picked) travel time. The distance between the rays at this depth is defined to be x_{err}. This objective function is reasonably stable in the presence of errors in the measured ray parameters.

assumed to be an exact parameter, rays would have to be traced and retraced until a pair of rays were found that intersected at a point where the travel time predicted by the ray tracing exactly matched the measured travel time.

The x-error objective function

Now I will describe the objective function I have chosen to use. This objective function is not overly sensitive to ray-parameter errors, nor is it expensive to compute.

Suppose that the upgoing and downgoing rays have both been traced back to some depth z in the velocity model (see Figure 7). Since the starting positions and the take-off angles (ray parameters) of both rays are known, the raypaths are unique. The *predicted travel time* at depth z can be defined as the computed travel time along the downgoing ray plus the computed travel time along the upgoing ray, as measured at depth z. Then for each pair of rays one can determine a depth z_e, the depth where the predicted travel time equals the observed travel time. Once the value of z_e has been determined, the horizontal distance between the two rays at that depth can be measured; this distance is defined as x_{err}. The objective function to be minimized is x_{err}^2. The value of x_{err} is clearly zero if the ray parameters and travel time have been measured correctly and if the velocity model has been chosen correctly. If the ray parameters p_s and p_g

have been measured incorrectly, x_{err} will be non-zero, but it won't vary wildly, even when the shot-geophone distance is small (compare this behavior to that of the objective function based on travel-time errors). And the rays need be computed only once per iteration; no expensive two-point ray-tracing algorithm is necessary.

In the problem I am solving, the velocity model is defined at regularly spaced grid points. In Appendix A I give the details of how ray tracing may be carried out in such a model.

MINIMIZING THE OBJECTIVE FUNCTION

Once an objective function has been defined, it is necessary to devise a method of minimizing it. To begin with, let us make some definitions.

First, a definition of the model: the model is defined to consist of N boxes. The velocity in box i is v_i (there may be a horizontal velocity gradient in each box, as well; see Appendix A). The vector $\mathbf{v}$ is made up of all N of the velocity terms v_i.

Next, a definition of the data: the data consist of M sets of picked parameters; the jth set of parameters contains $p_{s(j)}$, $p_{g(j)}$, $x_{s(j)}$, $x_{g(j)}$, and $t_{(j)}$.

Finally, a definition of the objective function: for the jth set of picked parameters, there is an associated error term $x_{\text{err}(j)}$. This term is found by ray-tracing methods as described in the previous section, it is clearly a function of $\mathbf{v}$, and it is conveniently written as $x_{\text{err}(j)}(\mathbf{v})$. The vector composed of all M of the error terms can be written as $\mathbf{x}_{\text{err}}$. The objective function Φ, then, can be written as

$$\Phi(\mathbf{v}) \equiv \sum_j x^2_{\text{err}(j)}(\mathbf{v}) = \|\mathbf{x}_{\text{err}}(\mathbf{v})\|^2. \tag{1}$$

It turns out to be necessary to include damping factors, which can prevent the velocity from varying too wildly in each iteration. Since the damping should be based on the overall velocity model, rather than on the change in velocity at a single iteration step (Tarantola and Valette, 1982), the damping factors can be incorporated into the objective function. As a result, the objective function becomes

$$\Phi(\mathbf{v}) \equiv \sum_j x^2_{\text{err}(j)} + \lambda_x^2 \sum_i \left(\frac{\partial v_i}{\partial x}\right)^2 + \lambda_z^2 \sum_i \left(\frac{\partial v_i}{\partial z}\right)^2, \tag{2}$$

where i is the index over model velocities, j is the index over picked parameters, and λ_x and λ_z are the horizontal and vertical damping terms, respectively.

The problem that is to be solved is now easily stated: find the value of $\mathbf{v}$ that minimizes $\Phi(\mathbf{v})$. This is a non-linear minimization problem; out of the many possible ways to solve such problems (Gill et al., 1981), I have chosen to use a modified Gauss-Newton method. Bishop et al. use a similar iterative method.

Let $\mathbf{v}^{(k)}$ be the current velocity model, the result of the kth iteration. The desired new velocity model is $\mathbf{v}^{(k+1)}$, which is to be found according to the formula

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \Delta\mathbf{v}. \tag{3}$$

In the standard Gauss-Newton method, $\Delta\mathbf{v}$ is found by solving the least-squares system

$$\underset{\sim}{\mathbf{A}}^{(k)}\Delta\mathbf{v} \approx -\mathbf{x}_{\text{err}}^{(k)}, \tag{4}$$

where $\mathbf{x}_{\text{err}}^{(k)} \equiv \mathbf{x}_{\text{err}}(\mathbf{v}^{(k)})$, $\underset{\sim}{\mathbf{A}}^{(k)}$ is the Fréchet matrix derived from $\mathbf{x}_{\text{err}}^{(k)}$, and $\Delta\mathbf{v}$ is the difference between the current velocity model and the desired model. The Fréchet matrix is defined according to the formula

$$A_{ij}^{(k)} \equiv \left(\frac{\partial x_{\text{err}(j)}}{\partial v_i}\right)_{\mathbf{v}=\mathbf{v}^{(k)}}. \tag{5}$$

This matrix can be obtained, for instance, through finite differencing methods: perturb each velocity term v_i slightly, and see how all the components of $\mathbf{x}_{\text{err}}$ vary in response. In practice, this method as described is too expensive; alternative methods are described in Appendix B (for laterally homogeneous velocity structures) and in Appendix C (for fully heterogeneous velocity structures).

There are a few details that must be added to this general description. Since the objective function contains a damping term (see equation 2), this term must be incorporated into the iterations. Define a differencing matrix $\underset{\sim}{\mathbf{D}}$ such that

$$\underset{\sim}{\mathbf{D}}\begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} \approx \left(\lambda_x\frac{\partial v_1}{\partial x},\ldots,\ \lambda_x\frac{\partial v_N}{\partial x},\ \lambda_z\frac{\partial v_1}{\partial z},\ldots,\ \lambda_z\frac{\partial v_N}{\partial z}\right)^T \tag{6}$$

The terms λ_x and λ_z have the same meaning as in equation (2). Now the least-squares system no longer takes the form of equation (4), but instead is written as

$$\begin{pmatrix} \underset{\sim}{\mathbf{A}} \\ \underset{\sim}{\mathbf{D}} \end{pmatrix}\Delta\mathbf{v} \approx \begin{pmatrix} -\mathbf{x}_{\text{err}} \\ -\underset{\sim}{\mathbf{D}}\mathbf{v} \end{pmatrix}. \tag{7}$$

As long as λ_x and λ_z are non-zero, equation (7) represents a sparse over-determined system of linear equations, even if M is less than N (as is the case with my synthetic examples). Equation (7) can be solved numerically by the conjugate gradient algorithm LSQR (Paige and Saunders, 1982).

An additional complication is that equation (3) should actually be written in the form

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \lambda\Delta\mathbf{v}, \tag{8}$$

where λ is the scalar which minimizes $\Phi(\mathbf{v}^{(k)} + \lambda\Delta\mathbf{v})$. If $\mathbf{v}^{(k)}$ is not too far from the value of $\mathbf{v}$ which minimizes Φ, λ will be close to 1.0. In other cases, however, setting λ equal to 1.0 may not

produce good results. In practice, it is very expensive to find λ exactly, since ray tracing must be performed every time Φ is calculated. It is relatively cheap, however, to find an approximate value of λ, if $\Phi(\mathbf{v}^{(k)})$, $\Phi(\mathbf{v}^{(k+1?)})$, $\underset{\sim}{\mathbf{A}}^{(k)}$, and $\underset{\sim}{\mathbf{A}}^{(k+1?)}$ are known (here $\mathbf{v}^{(k+1?)}$ is a trial value for $\mathbf{v}^{(k+1)}$, found by setting λ equal to 1 in equation (8), and $\underset{\sim}{\mathbf{A}}^{(k+1?)}$ is the corresponding Jacobian matrix). Without going into details, I will note that given such information, an approximate value of λ can be found through the cubic fit line-search method (Luenberger, 1984, pp. 205-206).

Appendix B contains the gruesome details of how the Fréchet matrix $\underset{\sim}{\mathbf{A}}$, defined in equation (5), is calculated when the velocity model is laterally homogeneous. One interesting result is that A_{ij} in general depends only on $p_{s(j)}$, $p_{g(j)}$, v_i, and $v_{I(j)}$, where $v_{I(j)}$ is the velocity at layer $I(j)$, the layer containing the endpoints of the jth pair of rays. This result helps make the problem a little bit more linear, since it means that A_{ij} does not depend on v_l if $l \neq (i \text{ or } I(j))$.

Finding the Fréchet matrix is a more difficult task when v is a function of both x and z. Besides the expected difficulties of computing derivatives in a layer containing a horizontal gradient, there is an additional problem. It has just been noted that in laterally homogeneous media, A_{ij} depends only on v_i and $v_{I(j)}$, where $I(j)$ is the index of the layer containing the endpoints of the jth ray. When the medium is laterally inhomogeneous, however, A_{ij} depends on the velocities in all of the boxes that the ray passes through on its way from the ith box to the endpoint. As a consequence, calculating A_{ij} is more complicated than for the flat-layer case. The details of how these calculations are carried out may be found in Appendix C.

RESULTS OF TESTS ON SYNTHETIC DATA

I have tested the CDR tomographic inversion method on synthetic data, including data from a vertically stratified (laterally homogeneous) model and from a fully inhomogeneous model. In all cases I obtained the picked parameters (x_s, x_g, p_s, p_g, t) directly from the modeling program, so I did not use the picking procedures described in the section on the CDR method. No noise was added to the synthetic data.

Synthetic data from a layered medium

The first two synthetic examples are from vertically stratified models.

The first model consisted of five layers, with a constant velocity within each layer. Figure 8 shows the raypaths that were generated during the modeling. The surface-position parameters, x_s and x_g, were rather arbitrary; they depended only on where the rays happened to intersect the surface. A total of 25 sets of CDR parameters were generated.

These synthetic data were used as input to the inversion algorithm described in the previous section. A vertically-stratified model was used in the inversion, with layers a distance of $\Delta z = .01$ apart; the total depth of the model was 1.0. The velocity boxes were each as wide as the model, so

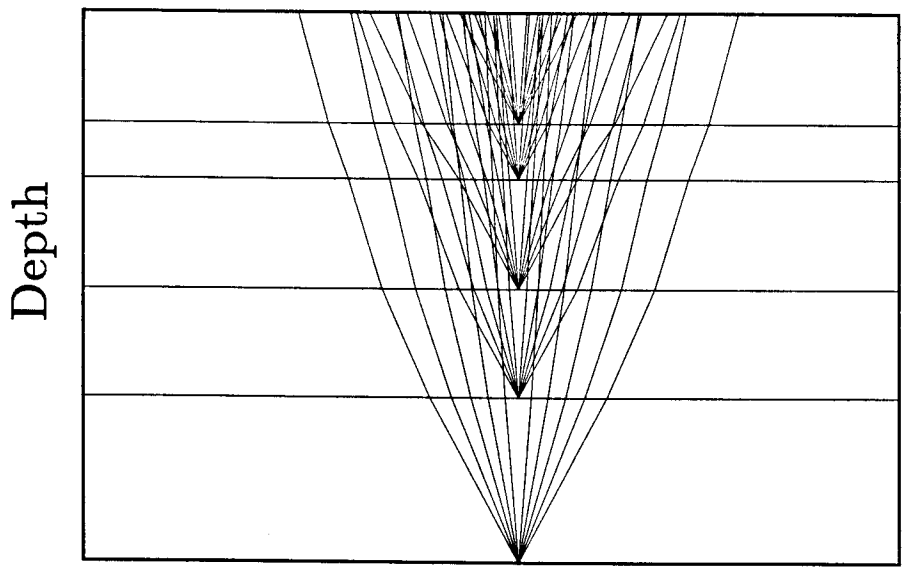


Figure 8: Generation of synthetic data. This model consists of 5 layers; velocity is constant within each layer. The third layer from the top is a low velocity zone. There is no vertical exaggeration.

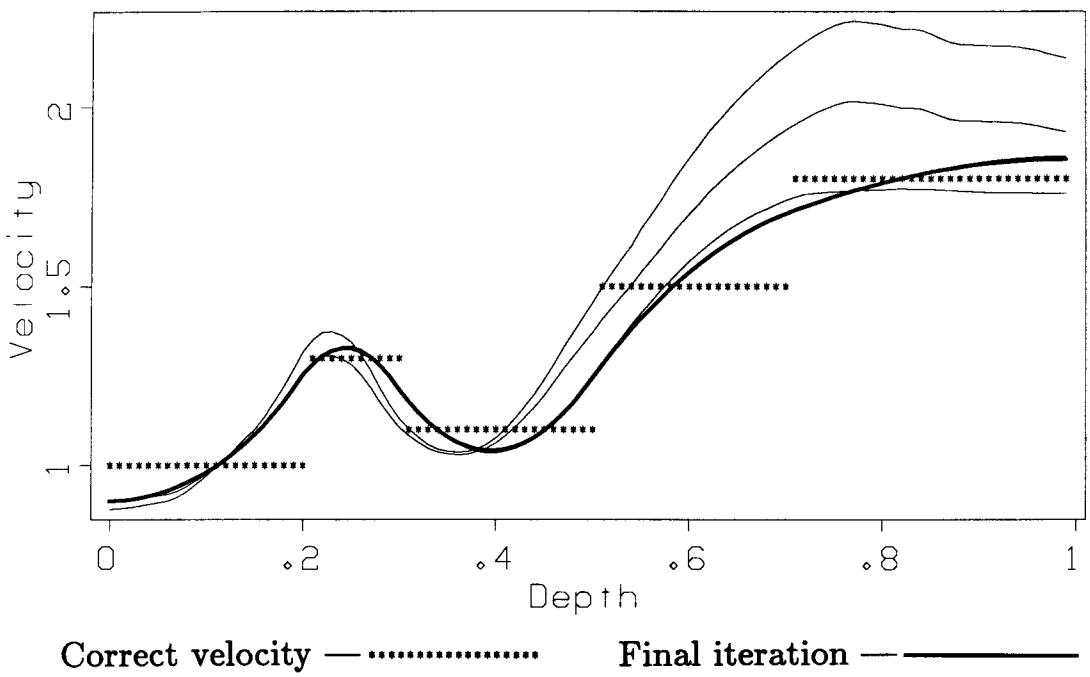


Figure 9: Iterative inversion. The synthetic data generated by the rays in Figure 8 were inverted by the method described in this paper. The results of the first 6 iterations are shown here; the result of the 6th iteration is drawn with a thicker line. The dotted line shows the correct velocity model.

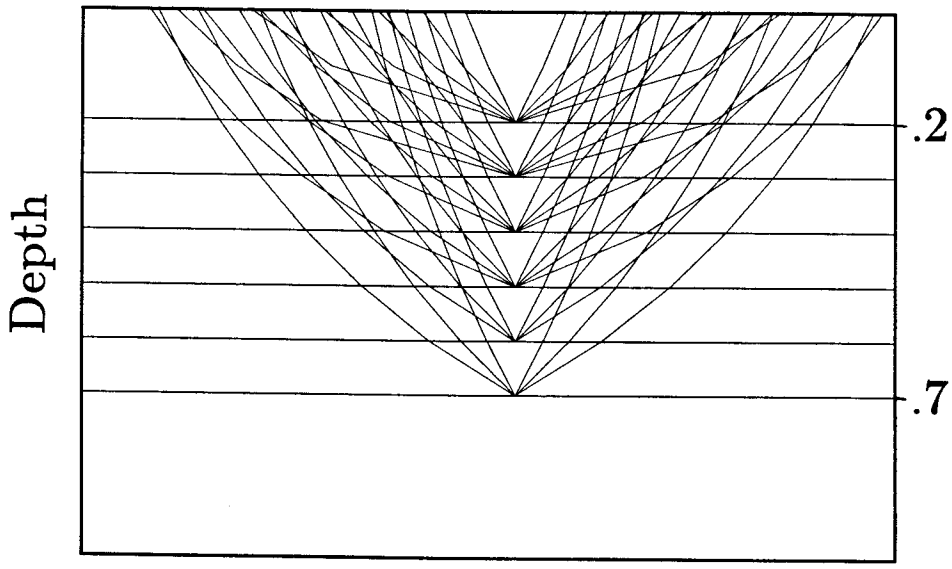


Figure 10: Generation of synthetic data. The velocity in the model varies linearly from a value of 1.0 at the top ($z = 0.$) to a value of 2.5 at the bottom ($z = 1.$). There are five reflecting boundaries within this model (although there are no sharp velocity changes), at the indicated depths. There is no vertical exaggeration.

that there was only one box per layer. The units used are unimportant, but the reader is free to choose any set of units that seems reasonable (distance in tens of kilometers and time in seconds, for instance). A vertical damping coefficient of $\lambda_z = .0015$ was used, and the initial velocity model was a constant velocity everywhere of 1.0. Figure 9 shows the results of the first 6 iterations on a plot of velocity versus depth. The results are very encouraging. The undamped inversion problem would be underdetermined, with only 25 sets of picked parameters and 100 layers in the model, so the damping term played an important role.

The second velocity model consisted of a linear vertical velocity gradient, with five embedded reflectors. Again, a vertically stratified model was used in the inversion, with layers a distance of $\Delta z = .01$ apart, and a total model depth of 1.0. A vertical damping coefficient of $\lambda_z = .001$ was used, and the initial velocity model was a constant velocity of 1.0. Figure 10 shows the raypaths generated during the modeling; the output consisted of 24 sets of CDR parameters.

Figure 11 shows the results of the inversion. The fit to the input model is good down to a depth of .7. There were no reflectors located below that depth, so deeper velocities could not be determined.

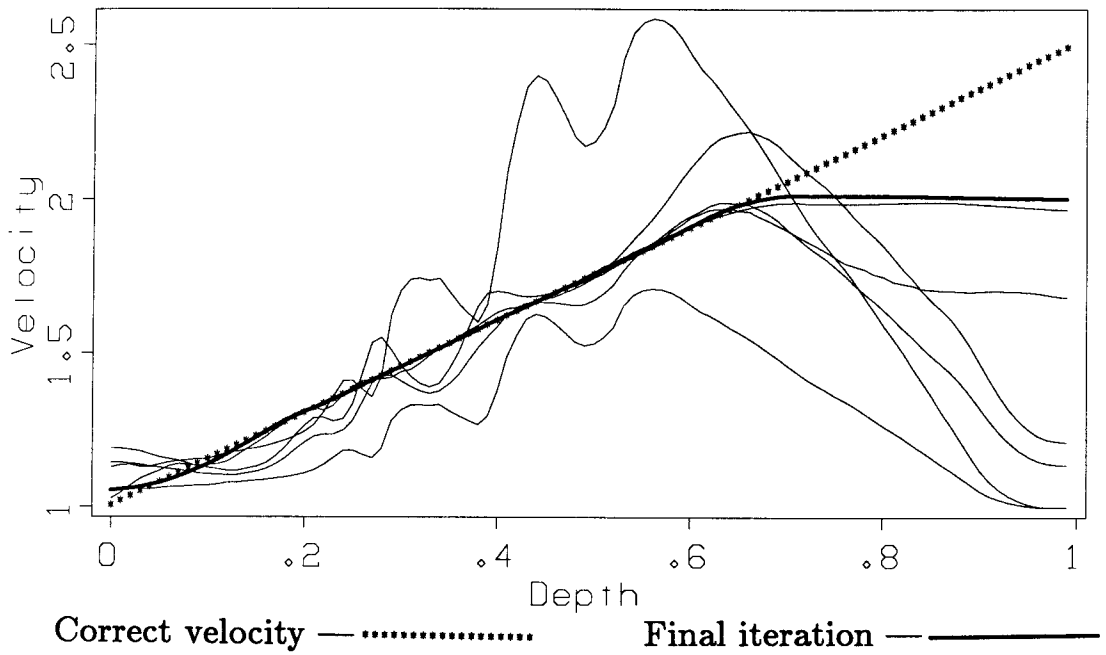


Figure 11: Iterative inversion. The picked parameters associated with the rays shown in Figure 10 were inverted, with the results shown here. The correct velocity model is shown by a dotted line, while the solid lines show the results of the first 7 iterations. The result of the 7th iteration is drawn with a thicker line.

Synthetic data from a laterally heterogeneous model

While tomographically inverting data from a vertically stratified medium is an interesting exercise, it isn't necessarily very practical, since there are so many cheaper ways of determining the velocity structure in such a medium. More interesting is to invert data from a medium that is both vertically and laterally inhomogeneous. I have chosen to invert synthetic data from a model similar to that used by Červený as a test of his ray-tracing program (see the title page of SEP-28). This model is shown in Figure 12, as are all the raypaths from a typical shot.

I used the ray-tracing program SEIS83 (Červený and Pšenčík, 1984) to generate the picked parameters. There were 19 geophones and 19 shots. It might be preferable to say that there were 19 shot arrays and 19 geophone arrays, since in practice the ray parameters p_s and p_g can be determined only by analyzing the data from arrays of adjacent shots and geophone. Any rays that arrived at the surface at too shallow an angle (more than 63.5 degrees from the vertical) were rejected. The output of the modeling program consisted of approximately 1400 sets of picked parameters.

The picked parameters were used as the input to my tomographic inversion program. The model used in the inversion consisted of a grid 10 km wide and 5.5 km deep, with a horizontal and vertical spacing of .05 km between adjacent grid points. Thus, there were 22,000 grid points

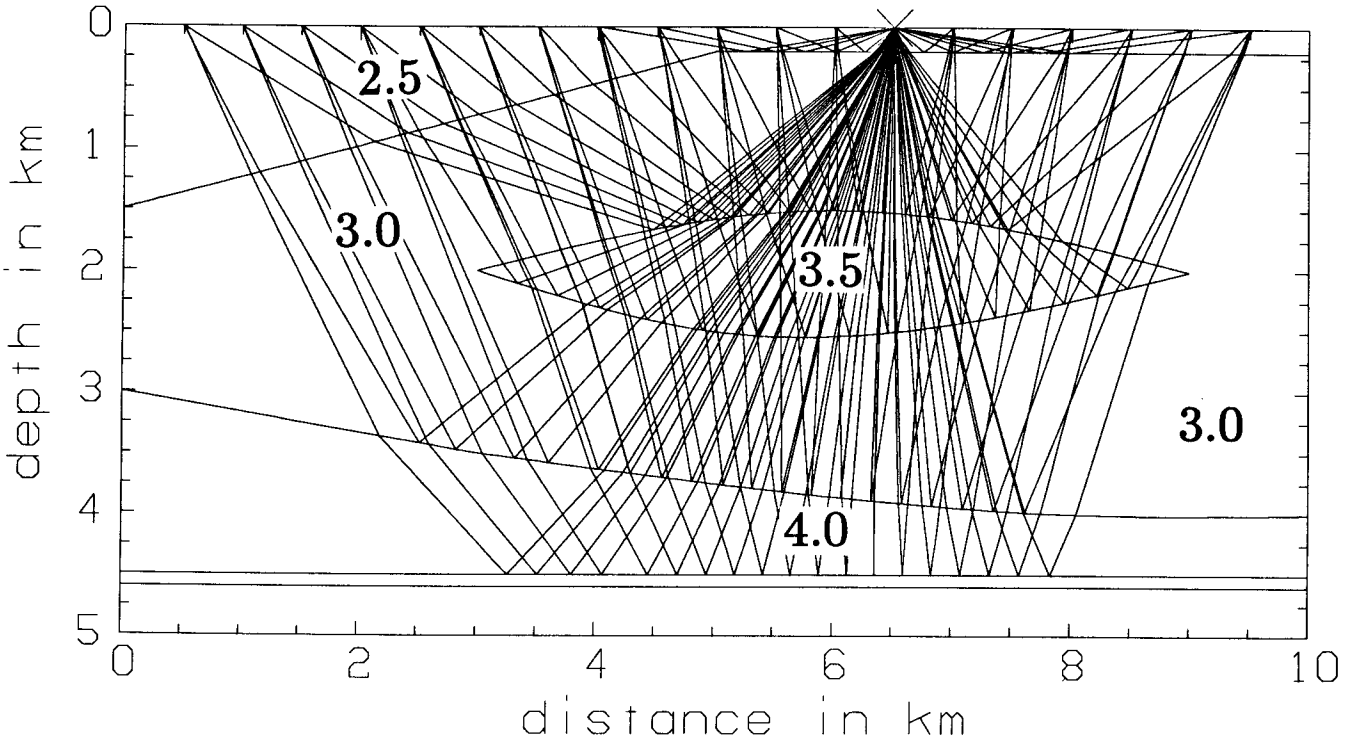


Figure 12: Synthetic model. This model was used to generate synthetic picked parameters. Note the depth of the deepest reflector. Shown are all the rays from a single shot; some of these rays were not used in the subsequent inversion, since they arrived at too shallow an angle. The shots and geophones were distributed along the surface, at intervals of .5 km, starting at the .5 km mark and going to the 9.5 km mark. Velocities are in km/sec.

in the inversion model. I tried two different initial velocity models (first guesses). The first one (which I will refer to as Model 3) varied linearly from 2.0 km/sec at the surface to 3.0 km/sec at the bottom of the model, while the second (which I will refer to as Model 4) varied linearly from 2.0 km/sec at the surface to 4.0 km/sec at the bottom. Note that although the bottom of the velocity model is 5.5 km deep, subsequent figures show velocities only to a depth of 5.0 km.

The results of inverting with Model 3 as an initial model are shown in Figure 13 (this Figure shows the results of the 18th iteration); the damping factors were $\lambda_x = .1$ and $\lambda_z = .05$. Agreement is good down to a depth of about 3 km, but below that depth the interval velocities are not very accurate. The problem is simply that these deeper velocities are not very strongly constrained by the data.

Figure 14 shows the results of inverting with Model 4 as an initial model (shown in this Figure are the results of the 15th iteration); the damping factors were $\lambda_x = .1$ and $\lambda_z = .025$. Note that this value of λ_z differs from the value used in the generation of Figure 13. Again, there is good agreement in the upper levels of the model, and now there is better agreement in the

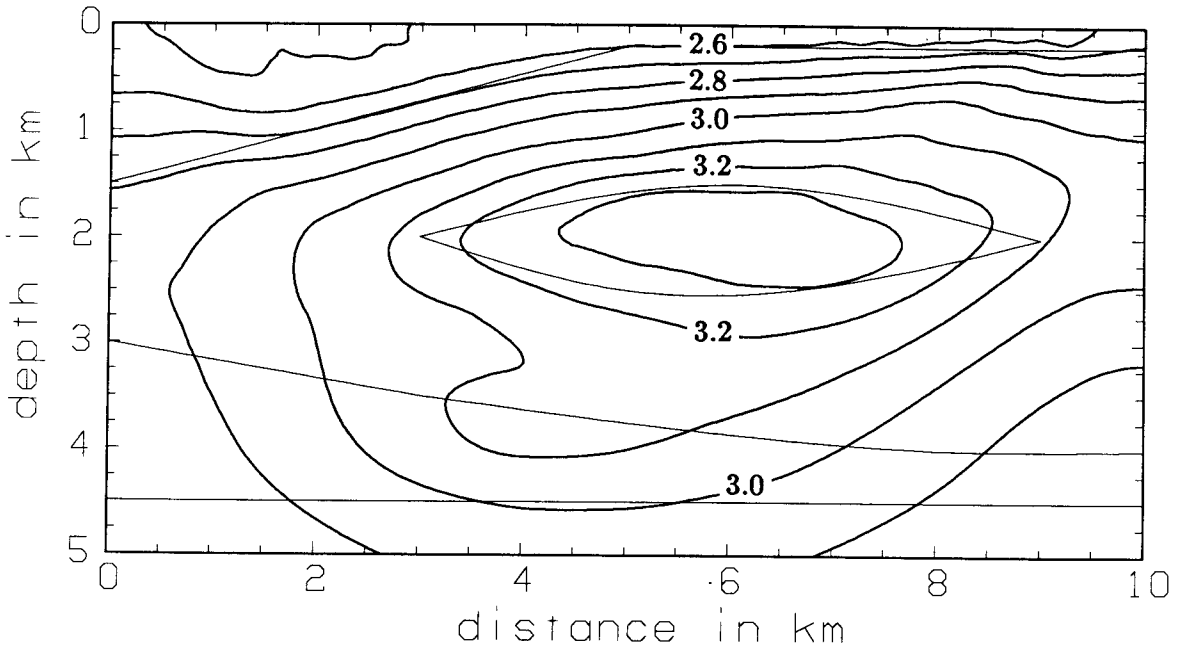


Figure 13: Inversion results (Model 3). This Figure shows the results of inverting the picked parameters from Figure 12, with Model 3 (see text) as the initial guess. The original model is overlaid for comparison; velocities are in km/sec.

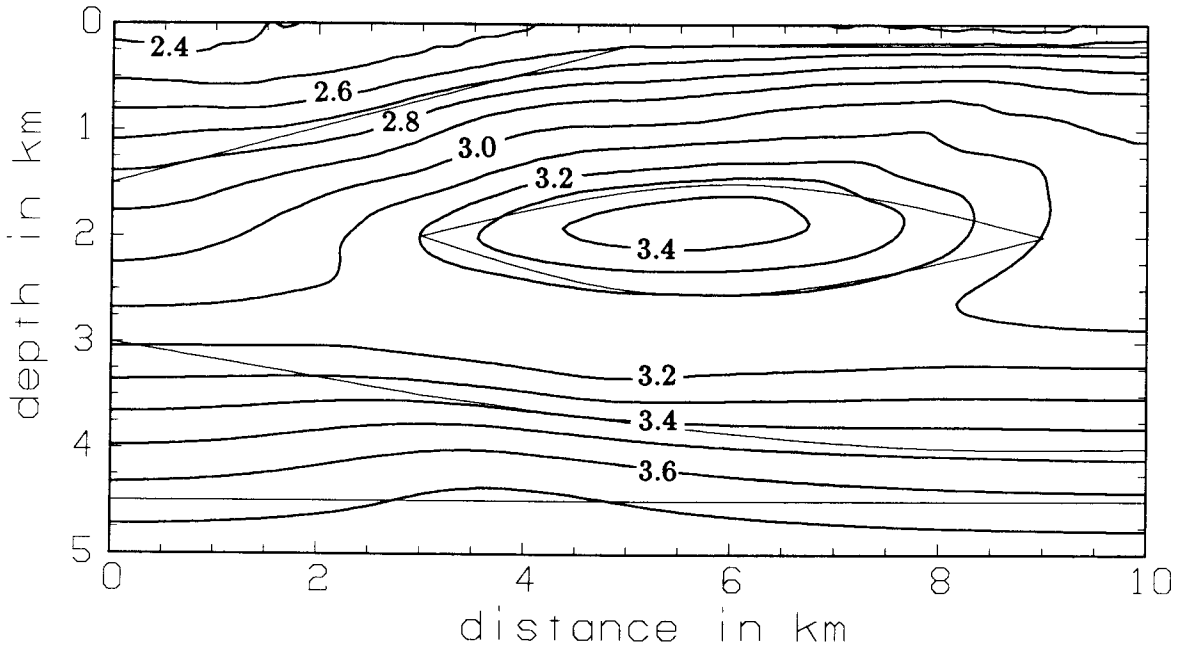


Figure 14: Inversion results (Model 4). This Figure shows the results of inverting the picked parameters from Figure 12, with Model 4 (see text) as the initial guess. The original model is overlaid for comparison; velocities are in km/sec.

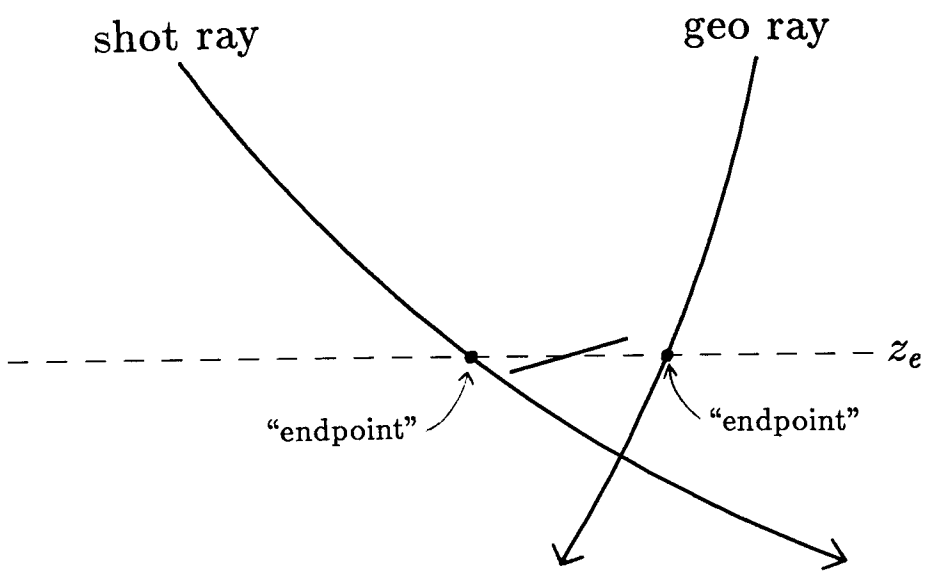


Figure 15: Basis of CDR migration. Given a set of picked parameters and a velocity structure, it is possible to draw a dip bar to represent a portion of the reflecting horizon that generated the rays. This dip bar is drawn at depth z_e, halfway between the endpoints of the shot and geophone rays. The dip of the dip bar corresponds to the dip of the reflector that would have generated those two rays.

lower levels as well. These two examples, when taken together, show how the final results can vary due to differences in the starting models. This sort of variability is not desirable, of course.

Once the velocity model were found by inversion, I performed CDR migration (Sword, 1984) on the picked parameters. In essence, I traced the rays corresponding to each set of picked parameters down to the appropriate depth z_e, and drew a dip bar at the midpoint between the endpoints of the shot and geophone rays (see Figure 15). The angle of the dip bar was based on the angle from the vertical of the two rays. In Figures 16 and 17 I show the the results of performing this process on the the velocity models shown in Figures 13 and 14, respectively. The width of each dip bar is equal to $(x_{\text{err}}/z_e)\times$ 1.0 km, so these Figures give some idea of how well the inversion process minimized x_{err}. Note that both velocity models were about equally successful in performing this minimization, which shows that the x_{err} objective function does not act as a strong constraint on interval velocities in the lower part of the model.

Most reflection seismologists (other than old-timers and adherents of CDR) are unaccustomed to seeing sections composed of dip bars, so I have rasterized and tapered the dip bars, and performed some time filtering, with the results shown in Figures 18 and 19. The widths of the dip bars are now based on travel times of the rays, rather than on x_{err}.

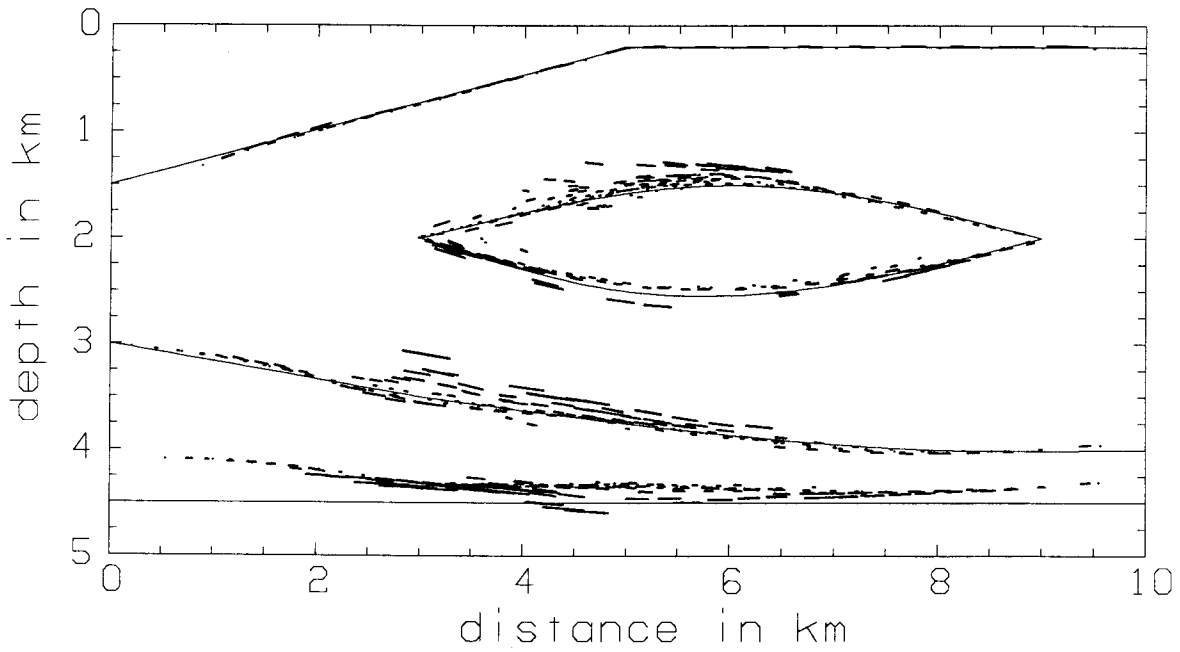


Figure 16: Results of CDR migration (Model 3). Based on the picked parameters generated from the model in Figure 12, and on the velocity structure shown in Figure 13, dip bars have been drawn according to the technique illustrated in Figure 15. The original model is overlaid for comparison.

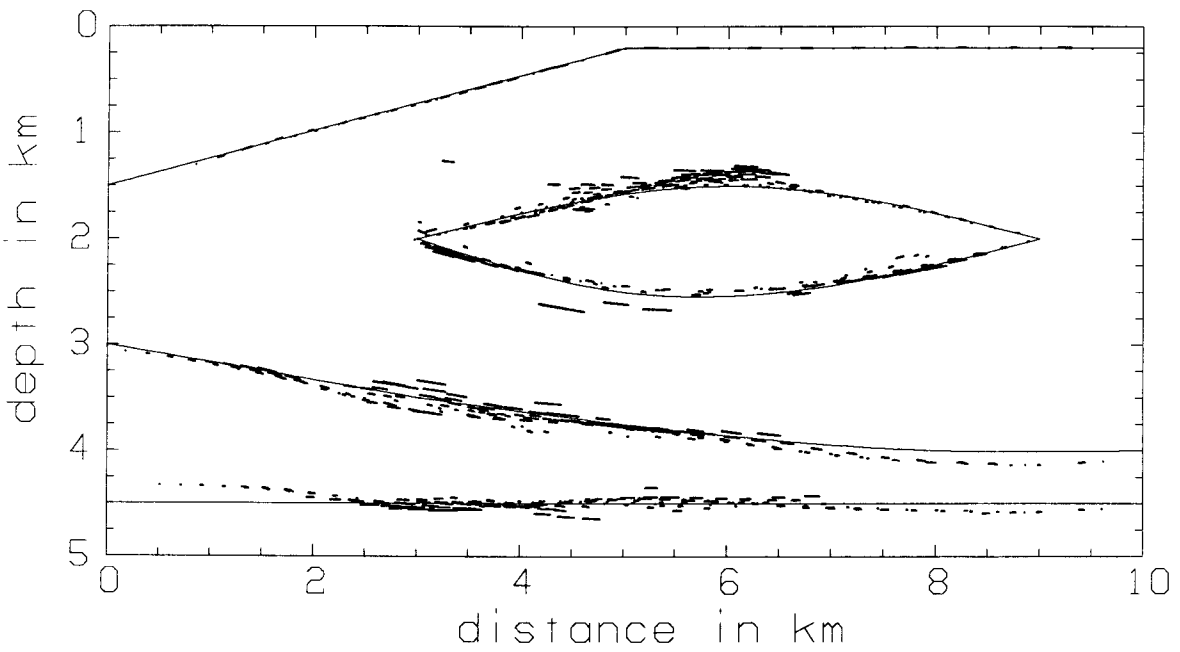


Figure 17: Results of CDR migration (Model 4). Based on the picked parameters generated from the model in Figure 12, and on the velocity structure shown in Figure 14, dip bars have been drawn according to the technique illustrated in Figure 15. The original model is overlaid for comparison.

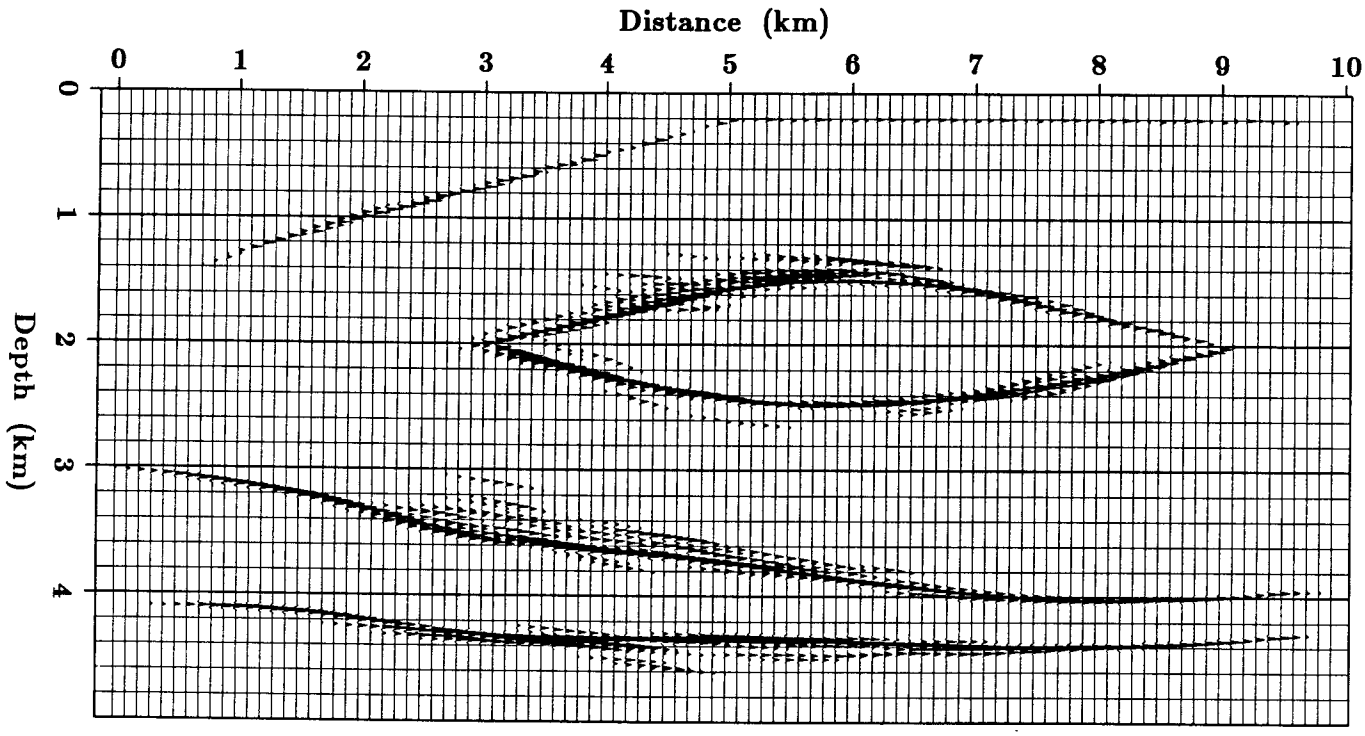


Figure 18: Filtered dip bars. The dip bars shown in Figure 16, after rasterization and filtering.

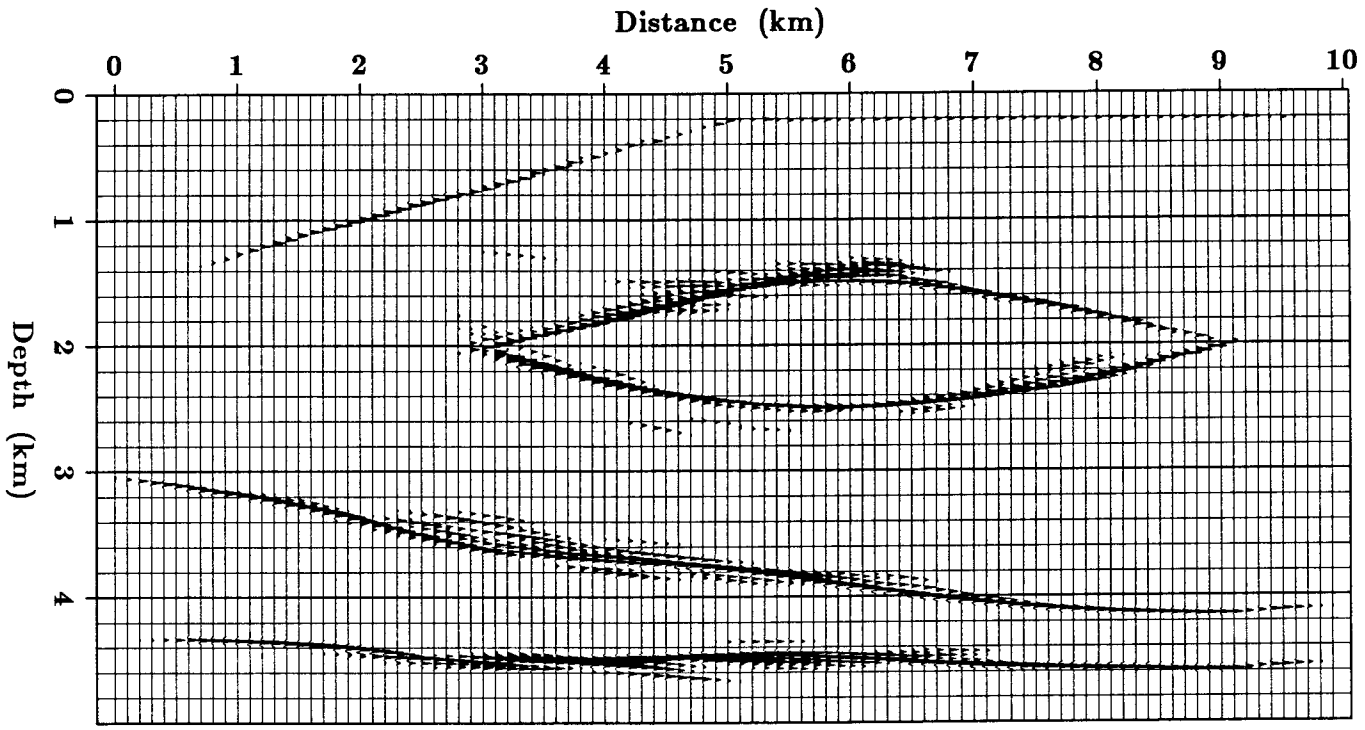


Figure 19: Filtered dip bars. The dip bars shown in Figure 17, after rasterization and filtering.

In this section, then, I have taken data from a synthetic model (Figure 12), inverted the data to find the velocity structure (Figure 14), and used this velocity structure to perform a pre-stack CDR migration on the data (Figure 19). Each velocity inversion typically required about 200 minutes of CPU time on the Convex C-1 (10 to 12 minutes per iteration), and the subsequent CDR pre-stack migration required less than 5 minutes.

DIRECTIONS OF FURTHER WORK

My results look promising so far, but there is more work to be done. Below I give some ideas that I should pursue.

Tests on real data

So far I have tested my inversion on noise-free synthetic data. I did not perform CDR analysis on the data in order to obtain the picked parameters; the modeling programs gave me these parameters directly. The inversion techniques should be tested on picked real data, or at least on noisy synthetic data. I should begin to think about how to discriminate against multiples and other forms of coherent noise.

Focusing of reflector images

The lower layers in Figures 16 and 17 are not very well focused. This is not too surprising; my inversion technique looks only at individual sets of picked parameters, without taking into account the possibility that several of these sets may have come from the same location of the same reflecting horizon. If I can come up with an objective function that measures the degree of focusing in my CDR migrated data, and if I can find a corresponding Fréchet matrix that relates the degree of focusing to changes in the velocity model, then perhaps I will have an inversion method that is better constrained than the present method. A better-constrained method might result in velocity models that are more accurate in their lower part.

Converted waves

Once a velocity model has been found that fits the PP reflected data, converted-wave data (from two-component geophones) can in principle be used to construct an S-wave velocity model. Converted wave data cannot be inverted by my method to give the P-wave and S-wave velocity fields simultaneously, but once one of these fields is found by using PP or SS reflection data, the converted-wave data can, in principle, be used to determine the other.

CONCLUSIONS

CDR tomographic inversion appears to be a useful method of determining interval velocities. I have demonstrated an algorithm that picks CDR parameters (p_s, p_g, x_s, x_g, and t) from conventionally recorded seismic data. I have shown that CDR tomographic inversion can invert noise-free synthetic data sets, and the results look promising, although the velocity in the lower part of the model is not as well determined as I would like.

IN MEMORIAM

Lev Aleksandrovich Riabinkin began his work in geophysics in 1930, and in 1934 participated in the first reflection seismology experiments carried out in the USSR. He initiated his research on the method of Controlled Directional Reception (CDR) in 1939, and in subsequent years developed it into a practical tool that was (and still is) widely used by Soviet exploration geophysicists. His text on the CDR method (Riabinkin et al., 1962) is the standard work in the field. He was always careful to point out that the basic ideas underlying CDR were first thought of by an American, Frank Rieber, but it is clear to the outside observer that Dr. Riabinkin deserves full credit for the subsequent development and application of CDR.

Lev Aleksandrovich Riabinkin was not only a leading researcher in Soviet reflection seismology, he was also a respected teacher. During his tenure as Chairman, the Field Geophysics Department of the Gubkin Institute graduated more than a thousand students, and two of his former graduate students are now themselves Department Chairmen at the Gubkin Institute.

As for his personal qualities, I will note only that Dr. Riabinkin was one of those rare individuals who truly embodied the term "a gentleman and a scholar". I consider myself fortunate that I was able to work for several months in his research group, and I, like all who knew him, shall miss him.

ACKNOWLEDGMENTS

This paper has benefited greatly from useful discussions: with Bill Harlan, Jeff Thorson, and Will Gray about the inversion of CDR data; with Stew Levin about non-linear inversion techniques; with Rick Ottolini about slant stacks; with Paul Fowler about all sorts of things; and with Dan Rothman and Ivan Pšenčík about dynamic ray tracing and the paraxial ray approximation. I would also like to thank Jon Claerbout for bringing the LSQR conjugate-gradient algorithm to my attention.

REFERENCES

Bishop, T.N., Bube, K.P., Cutler, R.T., Langan, R.T., Love, P.L., Resnick, J.R., Shuey, R.T., Spindler, D.A., and Wyld, H.W., 1985, Tomographic determination of velocity and depth in laterally varying media: Geophysics, **50**, 903–923.

Červený, V., and Pšenčík, I., 1984, Program SEIS83: Numerical modelling of seismic wave fields in 2-D laterally varying layered structures by the ray method (Fortran program).

Fowler, P., 1985, Migration velocity analysis by optimization: linear theory: SEP–44, 1–20.

Gill, P.E., Murray, W., and Wright, M.H., 1981, Practical optimization: Academic Press Inc.

Gray, W.C., and Golden, J.E., 1983, Velocity determination in a complex earth: 53rd Annual International SEG Meeting, September 15, Las Vegas, Nevada.

Harlan, W., and Burridge, R., 1983, A tomographic velocity inversion for unstacked data: SEP–37, 1–7.

Hermont, A.J., 1979, Letter to the Editor: Geophysics, **44**, 1601–1602.

Kong, S.M., Phinney, R.A., and Chowdhury, K.R., 1985, A nonlinear signal detector for enhancement of noisy seismic record sections: Geophysics, **50**, 539–550.

Levin, S.A., 1985, Newton trace balancing: SEP–42, 69–80.

Luenberger, D.G., 1984, Linear and nonlinear programming (2nd edition): Addison-Wesley Publishing Company Inc.

Ottolini, R., 1983, Signal/noise separation in dip space: SEP–37, 143-149.

Paige, C.C., and Saunders, M.A., 1982, LSQR: an algorithm for sparse linear equations and sparse least squares: ACM Transactions on Mathematical Software, **8**, 43-71.

Rapoport, M.B., 1977, Determination of wave parameters through the CDR summation of seismic traces, *in* Riabinkin, L.A., Ed., Digital processing of reflection seismic data: Transactions of the Gubkin Institute of Petrochemical and Gas Production (Moscow), **120**, 17–22 (in Russian).

Riabinkin, L.A., Napalkov, Yu.V., Znamenski, V.V., Voskresenski, Yu. N., and Rapoport, M.B., 1962, Theory and practice of the CDR seismic method: Transactions of the Gubkin Institute of Petrochemical and Gas Production (Moscow), **39** (in Russian).

Stoffa, P.L., Buhl, P., Diebold, J.B., and Wenzel, F., 1981, Direct mapping of seismic data to the domain of intercept time and ray parameter—A plane wave decomposition: Geophysics, **46**, 255–267.

Sword, C., 1984, The method of controlled directional reception: SEP–41, 369–393.

Tarantola, A., and Valette, B., 1982, Generalized nonlinear inverse problems solved using the least squares criterion: Reviews of Geophysics and Space Science, **20**, 219–232.

Toldi, J.L., 1985, Velocity analysis without picking: Ph.D. thesis, Stanford Univ. (also SEP–43).

Worthington, M.H., 1984, An introduction to geophysical tomography: First Break, **2**, 20–26.

APPENDIX A

RAY TRACING IN A GRIDDED VELOCITY FIELD

My inversion programs attempt to find the velocity field that best fits the picked seismic data. This velocity field can be expressed in the form of a grid, but the question then arises of how to trace rays through this grid. Here I present the methods that were used in producing the figures that appear in this paper.

Ray tracing in a laterally homogeneous medium

Ray tracing in a laterally homogeneous medium is relatively easy. If p is the ray parameter and z is depth, then

$$\sin\theta(z) = p\,v(z), \tag{A-1}$$

where $\theta(z)$ is the angle, measured from the vertical, of the ray at depth z, and $v(z)$ is the velocity of the medium. The ray parameter p is constant in a horizontally layered medium. If a particular layer has a thickness Δz, then the ray will travel within that layer a horizontal distance (defined as Δx) of

$$\Delta x = \Delta z \tan\theta(z). \tag{A-2}$$

Similarly, travel time (Δt) through that layer is

$$\Delta t = \frac{\Delta z}{v(z)\cos\theta(z)}. \tag{A-3}$$

It is assumed that velocity v (and thus θ as well) remains constant over the distance Δz. These three equations, plus some well-known trig identities ($\cos\theta = \sqrt{1-\sin^2\theta}$, for instance) are sufficient for finding x_{err}, given a velocity model and the parameters x_s, x_g, p_s, p_g, and t.

Ray tracing in a laterally heterogeneous medium

When velocity varies only vertically, so that $v = v(z)$, it is possible to discretize the velocity model in terms of layers. However, when velocity varies horizontally as well, I have found it preferable to discretize the model into boxes, allowing a horizontal velocity gradient to exist within each box. That is, if v_i is the velocity associated with a particular box, a_i is the value of the associated velocity gradient, and x_i is the horizontal position of the center of the box, then

$$v(x,z) = v_i + a_i(x - x_i) \tag{A-4}$$

within the box. The values of a_i are chosen so as to make $v(x,z)$ a continuous (but not continuously differentiable) function in x, even at the boundaries between horizontally adjacent boxes. Note,

however, that $v(x,z)$ will not be a continuous function in z, just as it was not a continuous function in z in the horizontally layered case.

Ray tracing is, of course, more complicated in the presence of a horizontal gradient. Suppose a ray, with ray parameter p_0, crosses into a new box at $x = 0$, $z = 0$. Let the velocity at that point be v_0, let the velocity within the box be described by the formula $v_0 + ax$, and define $k \equiv -a/v_0$. Then

$$\sin\theta = v_0 p_0, \tag{A-5}$$

where θ is the angle of the ray from the vertical immediately *after* it has entered the box. Suppose the ray emerges from the box at the point $x = \Delta x$, $z = \Delta z$, and that just before it emerges, it has an angle from the vertical of $\theta + \Delta\theta$. Then the following two formulas (adapted from Bishop et al.) hold:

$$\cos(\theta + \Delta\theta) = (1 - k\Delta x)\cos\theta, \tag{A-6}$$

and

$$\sin(\theta + \Delta\theta) = \sin\theta + k\Delta z\cos\theta. \tag{A-7}$$

Once $\sin\theta$ has been determined from equation (A-5), $\cos\theta$ may be obtained from a well-known trig identity, and Δz is known (it is the thickness of the layer, if the ray travels all the way from the top of the layer to the bottom). Then $\sin(\theta + \Delta\theta)$ may be obtained from equation (A-7), $\cos(\theta + \Delta\theta)$ obtained by a trig identity, and equation (A-6) solved to yield

$$\Delta x = \frac{1}{k}\left(1 - \frac{\cos(\theta + \Delta\theta)}{\cos\theta}\right). \tag{A-8}$$

Note that when k is close to zero, equation (A-2) should be used in place of this equation. If Δx is known, and Δz is the unknown quantity (this could happen if the ray intersects the vertical boundary between two horizontally adjacent boxes), equations (A-6) and (A-7) can be solved to yield a similar formula for Δz as a function of Δx. I will not give it here.

Another important parameter that must be determined is p_1, the ray parameter at the point of emergence. It is easily shown that $\sin(\theta+\Delta\theta) = (v_0+a\Delta x)p_1$, and this equation, in conjunction with equation (A-6) and the definition of k, can be solved to yield

$$p_1 = \frac{\cos\theta}{v_0}\tan(\theta + \Delta\theta). \tag{A-9}$$

Since velocity discontinuities occur only with changes in z (recall that the gradients a_i are chosen so as to make this statement true), p_1 remains unchanged as the ray crosses the interface into the next box, and it can be used as the input ray parameter, p_0, when the ray is traced through this next box.

It is worth noting that nowhere, so far, has it been necessary to evaluate a transcendental function. That is, the ray tracing can be carried out without the computer ever needing to evaluate

a sine, cosine, arcsine, or arccosine function. The only function (besides the four basic arithmetic functions) that needs to be called by this ray tracing program is the square root function (to solve such identities as $\sin^2\theta + \cos^2\theta = 1$). There is a single exception: the program needs to call the logarithm function to determine one necessary parameter.

This final parameter is Δt, the amount of time it take the ray to travel through a particular box. This quantity can be determined from the integral

$$\Delta t = \int_s \frac{ds}{v(s)}, \tag{A-10}$$

where s represents the path that the ray follows within the box. After a fair amount of algebra, this integral can be transformed into a more useful formula:

$$\Delta t = \frac{1}{kv} \log\left(\frac{\cos\theta\,(1+\sin(\theta+\Delta\theta))}{\cos(\theta+\Delta\theta)\,(1+\sin\theta)}\right). \tag{A-11}$$

When k is close to zero, it is preferable to use equation (A-3).

APPENDIX B

DETERMINING THE FRÉCHET MATRIX WHEN $v = v(z)$

Recall from equation (5) that $\underset{\sim}{\mathbf{A}}^k$, the Fréchet matrix at the kth iteration, is defined according to the formula

$$A_{ij}^k \equiv \left(\frac{\partial x_{\mathrm{err}(j)}}{\partial v_i} \right)_{\mathbf{v}=\mathbf{v}^k}. \tag{B-1}$$

Some more definitions are now necessary: Δz_i is the vertical distance that the jth ray travels in the ith layer (here, as in later definitions, the j subscript has been omitted for the sake of clarity); Δx_i is the horizontal distance that this ray travels in the ith layer; and Δt_i is the travel time of this ray within the layer. Then it is possible to write:

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial v_i} = \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} \cdot \frac{d\Delta x_i}{dv_i} + \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta t_i} \cdot \frac{d\Delta t_i}{dv_i}. \tag{B-2}$$

Now the problem is reduced to one of finding the value of each term.

First, however, there is a point that needs to be clarified. Recall that Δx_i has just been defined to be the horizontal distance traveled by the jth ray in the ith layer. This definition ignores the fact that two rays are being traced simultaneously in order to find $x_{\mathrm{err}(j)}$, and that $x_{\mathrm{err}(j)}$ is thus dependent on both of these rays. That is,

$$x_{\mathrm{err}} = x_{ge} - x_{se}, \tag{B-3}$$

where x_{se} is the horizontal position of the endpoint of the jth downgoing (shot) ray and x_{ge} is the horizontal position of the endpoint of the jth upgoing (geophone) ray (the endpoint of a ray is defined to be the point where the ray intersects the line $z = z_e$, with z_e defined as the depth where the combined computed travel times of the two rays equals the measured travel time). In practice, although a velocity change in a given layer will produce changes in both rays, it is most convenient to consider the effect of the change on only one of the rays at a time, and sum the results later into A_{ij}. This is the approach that will be used in this Appendix.

The first term in equation (B-2) is easily determined, after a moment of thought. Since v is a function only of z, x_{se} (or x_{ge}) varies exactly the same amount as Δx_i varies, independently of the velocity structure of the model. As a consequence of this result and of equation (B-3), and depending on which ray is under consideration,

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} = -1 \quad \text{(downgoing)}; \qquad \text{otherwise,} \quad \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} = 1 \quad \text{(upgoing)}. \tag{B-4}$$

The second term in equation (B-2) can be determined by differentiating equation (A-2). The result is:

$$\frac{d\Delta x_i}{dv_i} = \frac{p\Delta z_i}{\cos^3 \theta_i}, \tag{B-5}$$

where θ_i is the angle of the ray in layer i as measured from the vertical.

The third term in equation (B-2) is somewhat more complicated to derive. To begin with,

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta t_i} = \frac{dx_{\mathrm{err}}}{dz_e} \cdot \frac{dz_e}{d\Delta t_i}, \tag{B-6}$$

where z_e is the depth of the endpoints of the jth pair of rays. From equation (B-3),

$$\frac{dx_{\mathrm{err}}}{dz_e} = \frac{dx_{ge}}{dz_e} - \frac{dx_{se}}{dz_e}, \tag{B-7}$$

which through simple trigonometry becomes

$$\frac{dx_{\mathrm{err}}}{dz_e} = \tan\theta_{ge} - \tan\theta_{se}, \tag{B-8}$$

where θ_{ge} and θ_{se} are the angles at depth z_e of the upgoing and downgoing rays respectively. These angles can be derived from p_s and p_g through the rule that $vp = \sin\theta$.

The second part of equation (B-6) must be approached indirectly. First, note that even though travel time Δt_i within a layer may vary, the overall travel time t must remain constant. Thus, as Δt_i increases, $t - \Delta t_i$ decreases. As a result,

$$\frac{dz_e}{d\Delta t_i} = -\frac{dz_e}{dt}. \tag{B-9}$$

Next, it is useful to note that

$$\frac{dz_e}{dt} = \frac{1}{dt/dz_e}. \tag{B-10}$$

Then

$$\frac{dt}{dz_e} = \frac{dt_s}{dz_e} + \frac{dt_g}{dz_e}, \tag{B-11}$$

where t_s and t_g are the total travel times of the downgoing (shot) ray and the upgoing (geophone) ray, respectively. An application of the chain rule yields

$$\frac{dt}{dz_e} = \frac{dt_s}{dl_s}\frac{dl_s}{dz_e} + \frac{dt_g}{dl_g}\frac{dl_g}{dz_e}, \tag{B-12}$$

where l_s and l_g are the total path lengths of the downgoing (shot) ray and the upgoing (geophone) ray, respectively. Through the application of some trigonometry, and thanks to the fact that $v = dl/dt$,

$$\frac{dt}{dz_e} = \frac{1}{v_{se}}\frac{1}{\cos\theta_{se}} + \frac{1}{v_{ge}}\frac{1}{\cos\theta_{ge}}, \tag{B-13}$$

where $v_{se} \equiv v(x_{se}, z_e)$, and $v_{ge} \equiv v(x_{ge}, z_e)$.

Now it is possible to combine equations (B-6), (B-8), (B-9), (B-10), and (B-13) and write:

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta t_i} = -v_e \frac{\sin\theta_{ge}\cos\theta_{se} - \sin\theta_{se}\cos\theta_{ge}}{\cos\theta_{se} + \cos\theta_{ge}}, \tag{B-14}$$

where $v_e \equiv v(z_e)$.

The fourth term in equation (B-2) can be found by differentiating equation (A-3), with the result:

$$\frac{d\Delta t_i}{dv_i} = -\frac{\Delta t_i}{v_i} + \frac{p\Delta z_i \sin\theta_i}{v_i \cos^3\theta_i}. \tag{B-15}$$

The final result, found by substituting equations (B-4), (B-5), (B-14) and (B-15) into equations (B-1) and (B-2), is:

$$A_{ij}^k = \pm\frac{p\Delta z_i}{\cos^3\theta_i} - v_e\frac{\sin\theta_{ge}\cos\theta_{se} - \sin\theta_{se}\cos\theta_{ge}}{\cos\theta_{se} + \cos\theta_{ge}}\left(-\frac{\Delta t_i}{v_i} + \frac{p\Delta z_i \sin\theta_i}{v_i\cos^3\theta_i}\right). \tag{B-16}$$

The $\pm$ sign is positive when the ray under consideration is the upgoing (geophone) ray; otherwise it is negative.

APPENDIX C

DETERMINING THE FRÉCHET MATRIX WHEN $v = v(x, z)$

When the medium is both laterally and vertically heterogeneous, calculation of the Fréchet matrix is more complicated, but not unduly so.

Several different approaches to this calculation may be taken. The simplest approach is to perturb in turn each grid point in the velocity model, and see how all the x_{err} values vary. This approach is much too expensive, since the number of rays that would have to be traced equals the number of grid points in the model times the number of sets of picked parameters.

I have chosen to take a different approach, performing fewer perturbations and propagating the results of these perturbations by means of a transfer matrix. I shall begin by writing an equation analogous to equation (B-2):

$$\frac{\partial x_{\text{err}}}{\partial v_i} = \frac{\partial x_{\text{err}}}{\partial x_e} \cdot \frac{\partial x_e}{\partial v_i} + \frac{\partial x_{\text{err}}}{\partial t} \cdot \frac{\partial t}{\partial v_i}. \tag{C-1}$$

Here I am discussing the jth ray, but the subscript j has been dropped for the sake of clarity. I have defined v_i to be the velocity in the ith box of the model (technically, it is the velocity at the ith grid point, but it is more convenient to speak in terms of boxes), x_e is the horizontal position of the endpoint of the jth ray, and t is the travel time of the jth ray from the surface to the depth z_e. As in Appendix B, the endpoint of the ray is defined to be the point where the ray intersects the line $z = z_e$ (recall that z_e is the depth where the computed travel time equals the measured travel time). I also follow Appendix B in concentrating my attention on only one of the two rays associated with the jth set of picked parameters.

Equation (C-1) gives the overall formula for determining the Fréchet matrix; now it is necessary to find the values of the individual terms of this formula.

The first term in equation (C-1) is determined by a formula similar to equation (B-4):

$$\frac{\partial x_{\text{err}}}{\partial x_e} = -1 \quad \text{(downgoing)}; \qquad \text{otherwise,} \quad \frac{\partial x_{\text{err}}}{\partial x_e} = 1 \quad \text{(upgoing)}. \tag{C-2}$$

The third term in equation (C-1) is analogous to equation (B-14), but without the assumption that $v_{se} = v_{ge}$:

$$\frac{\partial x_{\text{err}}}{\partial t} = -v_{se} v_{ge} \frac{\sin\theta_{ge}\cos\theta_{se} - \sin\theta_{se}\cos\theta_{ge}}{v_{se}\cos\theta_{se} + v_{ge}\cos\theta_{ge}}, \tag{C-3}$$

with the variables on the right-hand side defined as in Appendix B.

The second and fourth terms of equation (C-1) are not found so easily, but they can be determined nonetheless. To begin with, I shall define some more terms, which are illustrated in Figure C-1. Recall that up until now I have denoted velocity boxes with a subscript i, meaning the ith box in the velocity model. It is useful at this point to introduce a different subscript, (l),

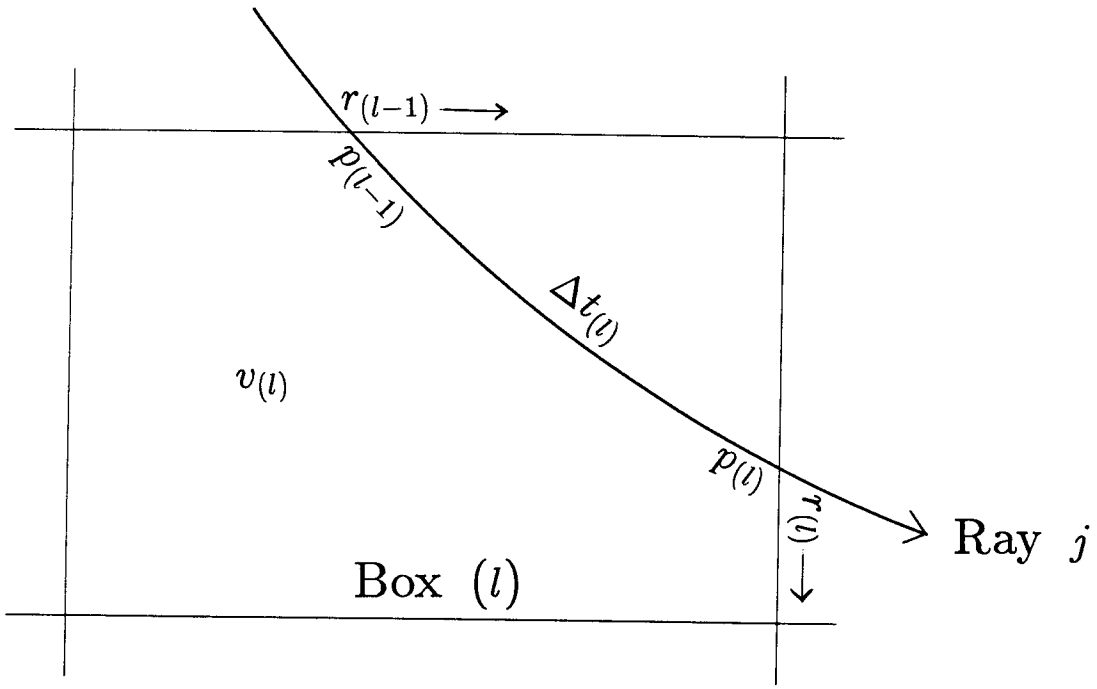


Figure C-1: Definition of terms. Ray j travels through box (l), the lth box along its path. It enters the box at spatial coordinate $r_{(l-1)}$ (measured parallel to the boundary where the ray enters), where it has ray parameter $p_{(l-1)}$. It leaves the box at $r_{(l)}$ (measured parallel to the boundary where the ray exits), with ray parameter $p_{(l)}$. The velocity within the box is $v_{(l)}$, and the ray takes an amount of time $\Delta t_{(l)}$ to travel through the box.

which means the lth box that ray j passes through (the parentheses are meant to distinguish this new type of subscript from the previous type). For instance, ray j starts at the surface, so the first box it passes through has velocity $v_{(1)}$, the second box it goes through has velocity $v_{(2)}$, and so on.

A ray entering the lth box has ray parameter $p_{(l-1)}$, and as it leaves the box it has ray parameter $p_{(l)}$. The position of the ray as it enters the box is denoted by $r_{(l-1)}$, which is a spatial coordinate measured along a certain axis; this axis is set to be parallel to the boundary that the ray crosses as it enters the box. The origin of this coordinate system is unspecified, since I will be concerned only with changes in $r_{(l-1)}$, not with its absolute value. Similarly, the position of the ray as it exits the box is denoted by $r_{(l)}$, which is measured along an axis parallel to the boundary that the ray crosses as it exits. This may seem like an odd way to measure the spatial coordinates of the ray, but it will make subsequent work easier.

The lth box has, as noted previously, a velocity $v_{(l)}$, and the ray takes an amount of time $\Delta t_{(l)}$ to travel through this box.

The box containing the endpoint of the ray may be denoted by L; by definition $r_{(L)}$ is measured along an axis parallel to the line $z = z_e$.

Now that these terms have been defined, it is possible to specify a transfer matrix $\underset{\sim}{\mathbf{T}}_{(l)}$:

$$\underset{\sim}{\mathbf{T}}_{(l)} \equiv \begin{pmatrix} \dfrac{\partial r_{(l)}}{\partial r_{(l-1)}} & \dfrac{\partial r_{(l)}}{\partial p_{(l-1)}} & 0 \\ \dfrac{\partial p_{(l)}}{\partial r_{(l-1)}} & \dfrac{\partial p_{(l)}}{\partial p_{(l-1)}} & 0 \\ \dfrac{\partial \Delta t_{(l)}}{\partial r_{(l-1)}} & \dfrac{\partial \Delta t_{(l)}}{\partial p_{(l-1)}} & 1 \end{pmatrix}. \tag{C-4}$$

The values of the terms in $\underset{\sim}{\mathbf{T}}_{(l)}$ may be determined analytically, but I prefer a finite-difference method: perturb $r_{(l-1)}$ and $p_{(l-1)}$, and see how $r_{(l)}$, $p_{(l)}$, and $\Delta t_{(l)}$ vary in response.

The next formula is essential, and I give it without proof:

$$\begin{pmatrix} \dfrac{\partial r_{(L)}}{\partial v_{(l)}} \\ \dfrac{\partial p_{(L)}}{\partial v_{(l)}} \\ \dfrac{\partial t}{\partial v_{(l)}} \end{pmatrix} = \underset{\sim}{\mathbf{T}}_{(L)} \cdot \underset{\sim}{\mathbf{T}}_{(L-1)} \cdots \underset{\sim}{\mathbf{T}}_{(l+2)} \cdot \underset{\sim}{\mathbf{T}}_{(l+1)} \cdot \begin{pmatrix} \dfrac{\partial r_{(l)}}{\partial v_{(l)}} \\ \dfrac{\partial p_{(l)}}{\partial v_{(l)}} \\ \dfrac{\partial \Delta t_{(l)}}{\partial v_{(l)}} \end{pmatrix}. \tag{C-5}$$

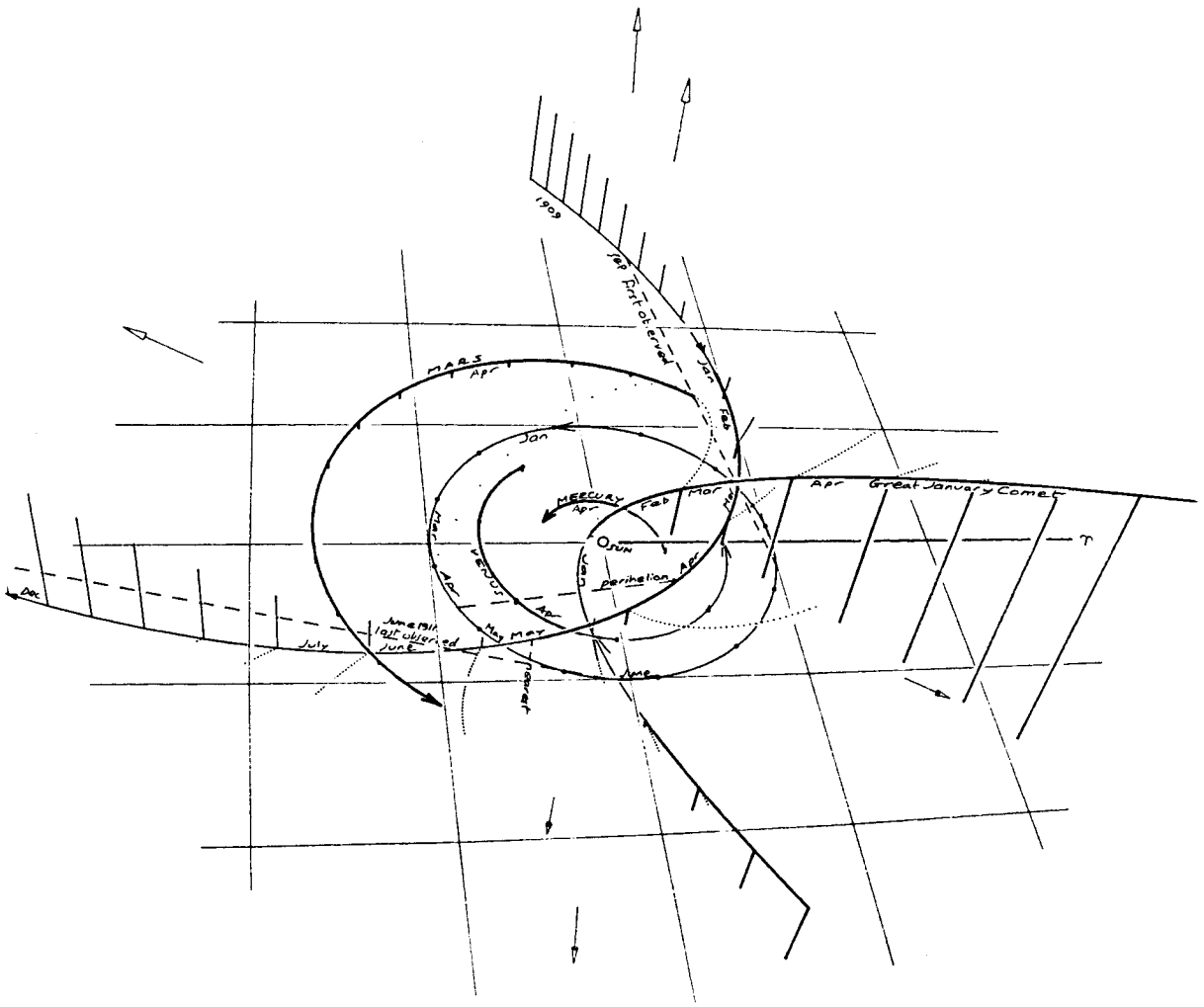
Be sure to distinguish between L and l in this formula. The quantities in the far right-hand vector ($\dfrac{\partial r_{(l)}}{\partial v_{(l)}}$, etc.) can be determined analytically, but again I prefer to find them through a finite-differencing method, perturbing $v_{(l)}$ and seeing the response.

Equation (C-5) gives the second and fourth terms of equation (C-1) directly, if the subscripts i and (l) refer to the same box (that is, if the ith box in the velocity model is also the lth box that ray j passes through). It is only necessary to make the identifications

$$\frac{\partial x_e}{\partial v_i} = \frac{\partial r_{(L)}}{\partial v_{(l)}} \quad \text{and} \quad \frac{\partial t}{\partial v_i} = \frac{\partial t}{\partial v_{(l)}}. \tag{C-6}$$

Halley's comet, 1910:
One of the best apparitions in history. The orbits of the comet and of earth pass close in two places: the second of these is better, as the comet is then well-baked from its recent passage by the sun and is very bright and has a long tail. In 1910 nearly the most favorable conjunction at this second approach point was achieved. Furthermore, the comet passed almost centrally across the face of the sun (a remarkable coincidence) and the fringes of its tail swept across the earth. The comet's solid nucleus, only a few miles across, passed across the sun unobserved, despite the best efforts of the day.

The great January comet snuck up on astronomers from behind the sun before any of them could attach their name, and burst upon the scene in full glory, brighter that even Halley's in that year. Many people, waiting for Halley's due a few weeks later, thought this WAS Halley's and never got the two straight.



Reprinted with Permission from Mankind's Comet, by Guy Ottewell and Fred Schaaf